SPARSE CODING WITH GATED LEARNED ISTA

Anonymous authors
Paper under double-blind review

ABSTRACT

In this paper, we study the learned iterative shrinkage thresholding algorithm (LISTA) for solving sparse coding problems. Following assumptions made by prior works, we first discover that the code components in its estimations may be lower than expected, i.e., require gains, and to address this problem, a gated mechanism amenable to theoretical analysis is then introduced. Specific design of the gates is inspired by convergence analyses of the mechanism and hence its effectiveness can be formally guaranteed. In addition to the gain gates, we further introduce overshoot gates for compensating insufficient step size in LISTA. Extensive empirical results confirm our theoretical findings and verify the effectiveness of our method.

1 INTRODUCTION

Sparse coding serves as the foundation of many machine learning applications, e.g., the direction-of-arrival estimation, signal denoising (Elad & Aharon, 2006), and super resolution imaging (Yang et al., 2010). In general, it aims to recover an inherently sparse vector $x_s \in \mathbb{R}^n$ from an observation $y \in \mathbb{R}^m$ corrupted by a noise vector $\varepsilon \in \mathbb{R}^m$. That is,

$$y = Ax_s + \varepsilon, \tag{1}$$

in which $A \in \mathbb{R}^{m \times n}$ is an over-complete basis matrix. The problem of recovering x_s, however, is a challenging task, in which the main difficulties are to incorporate the sparse constraint which is non-convex and to further determine the indices of its non-zero elements, i.e., the support of the vector. A reasonable solution to the problem is to use smooth and convex functions as surrogates to relax the constraint of sparsity, among which the most classical one probably is the l_1-norm penalty. Such a problem is carefully studied in Lasso (Tibshirani, 1996), and it can be solved via least angle regression (Efron et al., 2004), the iterative shrinkage and thresholding algorithm (ISTA) (Daubechies et al., 2004), etc.

Despite the simplicity, these conventional solvers suffer from critical shortcomings. Taking ISTA as an example, we know that 1) it converges very slowly with only a sublinear rate (Beck & Teboulle, 2009), 2) the correlation between each of the two columns of A should be relatively low. In recent years, deep learning (LeCun et al., 2015) methods have achieved remarkable successes. Deep neural networks (DNNs) have been proven both effective and efficient in dealing with many tasks, including image classification (He et al., 2016), object detection (Girshick, 2015), speech recognition (Hinton et al., 2012), and also sparse coding (Gregor & LeCun, 2010; Wang et al., 2016; Borgerding et al., 2017; He et al., 2017; Zhang & Ghanem, 2018; Chen et al., 2018; Liu et al., 2018; Sulam et al., 2019). The core idea behind deep learning-based sparse coding is to train DNNs to approximate the optimal sparse code. For instance, an initial work of Gregor and LeCun's (2010) takes the inspiration from ISTA and develops an approximator named learned ISTA (LISTA), which is structurally similar to a recurrent neural network (RNN).

It has been demonstrated both empirically and theoretically that LISTA is superior to ISTA (Wang et al., 2016; Moreau & Bruna; Giryes et al., 2018; Chen et al., 2018). Nevertheless, it is also uncontroversial that there exists much room for further enhancing it. In this paper, we delve deeply into the foundation of (L)ISTA and discover possible weaknesses of LISTA. First and foremost, we know from prior arts (Chen et al., 2018; Liu et al., 2018) that LISTA tends to learn large enough biases to achieve no "false positive" in the support of generated codes and further ensure linear convergence, and we prove that this tendency, however, also makes the magnitude of the code components being lower than that of the ground-truth. That said, there probably exists a requirement of *gains* in the code estimations. Second, regarding the optimization procedure of ISTA as to minimize an upper bound of its objective function at each step, we conjecture that the element-wise update of (L)ISTA normally "lags behind" the optimal solution, which suggests that it requires *overshoots* to reach the optimum, just like what has been suggested in fast ISTA (FISTA) (Beck & Teboulle, 2009) and learned FISTA (LFISTA) (Moreau & Bruna).

In this paper, our main contributions are summarized as follows:

- We discover weaknesses of LISTA by theoretically analyzing its optimization procedure, for mitigating which we introduce gain gates and overshoot gates, akin to update gate and reset gate mechanisms in the gated recurrent unit (GRU) Cho et al. (2014).
- We provide convergence analyses for LISTA (with or without gates), which further give rise to conditions on which the performance of our method with gain gates can be guaranteed. A practical case is considered, where the assumption of no "false positive" is relaxed.
- Insightful expressions for the gates are presented. In comparison with state-of-the-art sparse coding networks (not limited to previous extensions to LISTA), our method achieves superior performance. It also applies to variants of LISTA, e.g., LFSITA (Moreau & Bruna) and ALISTA (Liu et al., 2018).

Notations: In this paper, unless otherwise clarified, vectors and matrices are denoted by lowercase and uppercase characters, respectively. For vectors/matrices originally introduced without any subscript, adding a subscript (e.g., i) indicates its element/column at the corresponding position. For instance, for $x \in \mathbb{R}^n$, x_i represents the i-th element of the vector, and $W_{:,i}$ denotes the i-th column of a matrix $W \in \mathbb{R}^{n\times n}$. While for vectors introduced with subscripts already, e.g., x_s, we use $(x_s)_i$ to denote its i-th element. The operator $\odot$ is used to indicate element-wise multiplication of two vectors. The support of a vector is denoted as $\text{supp}(x) := \{i | x_i \neq 0\}$. We use $\sup_{x_s}$ as the simplified form of $\sup_{x_s\in\mathcal{X}(B,s,0)}$, see Assumption 1 for the definition of $\mathcal{X}(B, s, 0)$.

2 Background

In general, sparse coding solves the problem that can be formulated as

$$\min_x f(x, y) + \lambda r(x), \tag{2}$$

in which $f(x, y)$ calculates the residual of approximating y using a linear combination of column-wise features in A. The function $f(x, y)$ is convex with respect to x in general. In particular, if ε is a Gaussian vector, then it should be $f(x, y) = \|Ax - y\|_2^2$. The term $\lambda r(x)$ serves as a regularizer for sparsity and we have $r(x) = \|x\|_1$ in Lasso. As mentioned, a variety of algorithms can be applied to solve the problem and our focus in the paper is (L)ISTA. We first revisit the optimization procedure of ISTA, which is the foundation of LISTA as well. Given y, let us introduce a scalar $\gamma > 0$ that fulfills $\gamma I - \nabla_x^2 f(x, y) \succ 0, \forall x$, then it can be considered as optimizing an upper bound of the objective function obtained via Taylor expansion. To be more specific, for any presumed $x^{(t)}$, we have

$$f(x, y) + \lambda r(x) \leq f(x^{(t)}, y) + (x - x^{(t)})\nabla_x f(x^{(t)}) + \frac{\gamma}{2}\|x - x^{(t)}\|^2 + \lambda r(x). \tag{3}$$

By substituting $r(x)$ with $\|x\|_1$ and optimizing the bound in an element-wise manner, we can easily get the one-step update that zeros the gradient based on $x^{(t)}$. It is, $x^{(0)} = \mathbf{0}$ and

$$x^{(t+1)} = s_{\lambda/\gamma}(x^{(t)} - \nabla_x f(x^{(t)})/\gamma), \quad \forall t \geq 0, \tag{4}$$

in which $s_b(x) := \text{sign}(x)(|x| - b)_+$ is a shrinking function and $(\cdot)_+$ is a rectified linear unit (ReLU) calculating $\max\{0, \cdot\}$. For Gaussian noises, the formulation reduces to

$$x^{(t+1)} = s_{\lambda/\gamma}\left(\left(I - \frac{A^T A}{\gamma}\right) x^{(t)} + \frac{A^T}{\gamma} y\right). \tag{5}$$

The update as shown in Eq. (4) and (5) can be performed iteratively until convergence. However, the convergence of ISTA (along with some other conventional solvers) is known to be slow, and it has been shown that DNNs can be utilized to accelerate the procedure. Many researchers have explored the idea since the initial work of Gregor and LeCun's (i.e., LISTA). For LISTA, they design deep architectures following the main procedure of ISTA yet to learn parameters in an end-to-end manner from data (Gregor & LeCun, 2010; Hershey et al., 2014). The inference process of LISTA is similar to that of a RNN and can be formulated as $x^{(0)} = \mathbf{0}$ and

$$x^{(t+1)} = s_{b^{(t)}}(W^{(t)}x^{(t)} + U^{(t)}y), \quad t = 0, \cdots, d-1, \tag{6}$$

where $\Theta = \{U^{(t)}, W^{(t)}, b^{(t)}\}_{t=0,1,\dots,d-1}$, is learnable parameters set. Some works (Xin et al., 2016; Chen et al., 2018) have proved that $W^{(t)}$ and $U^{(t)}$ should satisfy the constraint $W^{(t)} = I - U^{(t)}A$, such that

$$x^{(t+1)} = s_{b^{(t)}}(x^{(t)} + U^{(t)}(Ax^{(t)} - y)), \quad t = 0, \cdots, d-1. \tag{7}$$

The parameters in Θ are normally learned from a set of training samples by minimizing the difference between the final code estimations and ground-truth. In this paper, our main assumption for theoretical analyses follows those of prior works (Chen et al., 2018; Liu et al., 2018) in a noiseless case, and noisy cases will be considered in the experiments.

Assumption 1. *The sparse vector x_s and noise vector ε are sampled from a set $\mathcal{X}(B, s, 0)$ fulfilling:*

$$\mathcal{X}(B, s, 0) := \{x \big| \|x\|_\infty \le B, \|\varepsilon\|_\infty \le 0, \|x\|_0 \le s\}.$$

3 Sparse Coding with Gain Gates and Overshoot Gates

In this section, we will introduce the advocated gain gates and overshoot gates. Along with thorough discussions for the motivations, their formulations are provided in Section 3.1 and 3.2, respectively. Figure 1 summarizes the inference process of the standard LISTA and two evolved versions with our gates incorporated.

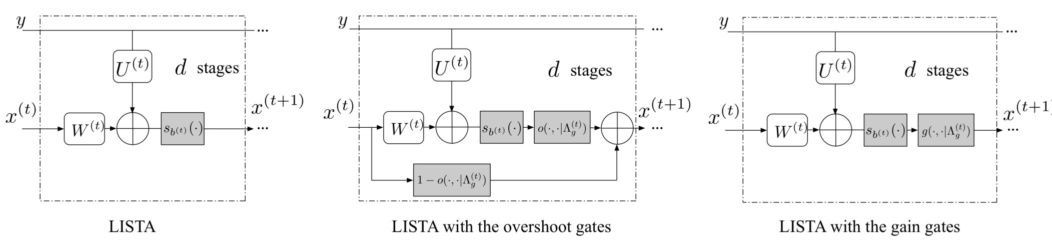


Figure 1: The inference process of the standard LISTA and evolved versions with our gates

3.1 Sparse Coding with Gain Gates

Recent works have shown linear convergence of LISTA (Chen et al., 2018; Liu et al., 2018). In order to guarantee the convergence, it is also demonstrated that the value of bias terms should be large enough to eliminate all "false positive" in the support of the generated codes. However, this may lead to an issue that the magnitude of the generated code components in LISTA must be smaller than those of the ground-truth. Our result in Proposition 1 makes this formal. For clarity of the result, we would like to introduce the following definition first.

Definition 1. *(Liu et al., 2018) Given a matrix $A \in \mathbb{R}^{m\times n}$, its generalized mutual coherence is:*

$$\mu(A) := \inf_{W\in\mathbb{R}^{n\times m}, W_{i,:}A_{:,i}=1,\forall i} \Big\{ \max_{i\neq j, 1\le i,j\le n} W_{i,:}A_{:,j} \Big\}. \tag{8}$$

We let $\mathcal{W}(A)$ denote a set of all matrices that can achieve the generalized mutual coherence $\mu(A)$, which means:

$$\mathcal{W}(A) := \Big\{ W \big| \max_{i\neq j, 1\le i,j\le n} W_{i,:}A_{:,j} = \mu(A), W_{i,:}A_{:,i} = 1, \forall i \Big\}. \tag{9}$$

Proposition 1. *(Requirement of gains). With $U^{(t)} \in \mathcal{W}(A)$ and $W^{(t)} = I - U^{(t)}A$, if $b^{(t)} = \mu(A)\sup_{x_s}\|x^{(t)} - x_s\|_1$ is achieved in LISTA to guarantee no "false positive" (i.e., supp$(x^{(t)}) \subset$ supp(x_s)) and further linear convergence (i.e., $\|x^{(t)} - x_s\|_2 \le sB\exp(ct)$, in which $c = \log((2s-1)\mu(A))$), then we have for the estimation $|x_i^{(t)}| \le |(x_s)_i|$ and $x_i^{(t)}(x_s)_i \ge 0, \forall i \in$ supp(x_s).*

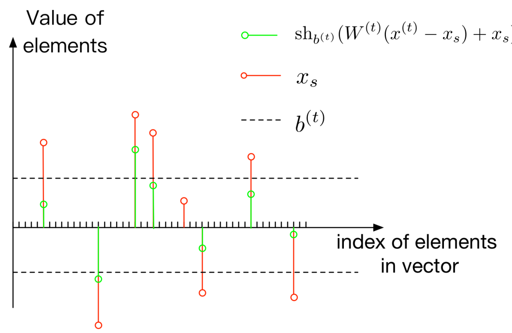


Figure 2: The generated code estimation can be more accurate if we enforce gains on its components.

Provided Proposition 1 as the evidence of a potential weakness of LISTA, we believe that if the code components can be enlarged appropriately, then the estimation at each step would be closer to x_s, and the convergence of LISTA will be further improved, which inspires us to design a gate to enlarge the generated code components. Such a gate is named as a gain gate and it acts on the input to the current estimation, akin to a reset gate in GRU (Cho et al., 2014), which is

$$x^{(t+1)} = s_{b^{(t)}}(W^{(t)}(g_t(x^{(t)}, y|\Lambda_g^{(t)}) \odot x^{(t)}) + U^{(t)}y), \tag{10}$$

in which the gate function $g_t(\cdot, \cdot|\Lambda_g^{(t)})$ outputs an n-dimensions vector. In the original implementation of LISTA, the output of each layer is obtained by calculating Eq. (4) iteratively. It has been proven that the estimation $x^{(t)}$ ultimately converges to the ground-truth x_s (as $t \to \infty$), only if the condition of $(W^{(t)} - (I - U^{(t)}A)) \to 0$ holds. That said, the learnable matrices $U^{(t)}$ and $W^{(t)}$ are suggested to be entangled to the end. Yet, with our gated mechanism, the update rule has been modified into Eq. (10), making it unclear whether the convergence is guaranteed similarly or not. To figure it out, we perform theoretical analyses in depth, which will further provide guidance for the gate design. We are going to explore: whether the learnable matrices are still entangled as in LISTA, and to encourage fast convergence, what properties should the gate function satisfy? Theorem 1 and 2 give some answers to these questions and they are based on the same assumptions as for Proposition 1.

Theorem 1. *If the s-th order principal subformula of $W^{(t)}$ have full rank, then for the gate function bounded from both above and below, we have x_s as the fixed point of Eq. (10) only if*

$$\mathrm{diag}(g_t(x^{(t)}, y|\Lambda_g^{(t)})) \to D \quad \textit{and} \quad W^{(t)}D - (I - U^{(t)}A) \to 0, \quad \textit{as } t \to \infty, \tag{11}$$

in which D is an $n \times n$ constant matrix and the function $\mathrm{diag}(\cdot)$ creates a diagonal matrix with the elements of its input on the main diagonal.

From Theorem 1 we can equivalently have $(\tilde{W}^{(t)} - (I - U^{(t)}A)) \to 0$ by defining $\tilde{W}^{(t)} := W^{(t)}D$, which means the learnable matrices are similarly entangled as in the standard LISTA. Besides, we know that as the number of layers increases, each introduced gain gate should ultimately converge to a constant (diagonal) matrix D to guarantee performance. This result may hint us to more specific gain gate functions. As we also know from Proposition 1 that gains greater than one are perhaps more appropriate, we advocate, for each index i of the vector,

$$g_t(x^{(t)}, y|\Lambda_g^{(t)})_i = 1 + \kappa_t(x^{(t)}, y|\Lambda_g^{(t)})_i \quad \text{and} \quad \kappa_t(x^{(t)}, y|\Lambda_g^{(t)})_i > 0, \tag{12}$$

in which $\kappa_t(x^{(t)}, y|\Lambda_g^{(t)})_i$ is the i-th element of $\kappa_t(x^{(t)}, y|\Lambda_g^{(t)})$ and it should decrease as t increases, in order to guarantee convergence in Eq. (11). We further study convergence rate of "LISTA" equipped with such gain gates. For clarity, let us introduce another condition for the function before moving to more details:

$$\kappa_t(x^{(t)}, y|\Lambda_g^{(t)})_i < 2b_i^{(t-1)}/|x_i^{(t)}|. \tag{13}$$

We present theoretical results as follows on the basis of Proposition 1, i.e., we still have $U^{(t)} \in \mathcal{W}(A)$, $W^{(t)} = I - U^{(t)}A$ and Assumption 1, but the requirement for $b^{(t)}$ is different.

Theorem 2. *If $b^{(t)} = \mu(A) \sup_{x_s} \|x_s - x^{(t)} \odot g_t(x^{(t)}, y|\Lambda_g^{(t)})\|_1$ is achieved, following the update rule in Eq. (10), if the conditions in Eq. (12) and (13) hold for the gate function, there will be*

$$\|x^{(t)} - x_s\|_2 \leq sB \exp(\sum_{i=1}^{t-1} c_i + c), \tag{14}$$

in which $c = \log((2s-1)\mu(A))$, $c_i = c$ if $i < \lceil \log(\frac{sB}{\|x_s\|_1}) / \log(\frac{1}{(2s-1)\mu(A)}) \rceil$, and $c_i < c$ otherwise.

Theorem 2 presents an upper bound of $\|x^{(t)} - x_s\|_2$ for LISTA with gain gates, and it shows that so long as the gates satisfying conditions in Eq. (12) and (13) are introduced, the "convergence factor" $c + \sum c_i$ of our gated LISTA would be smaller in comparison with that of the standard LISTA (which is ct, see Proposition 1 and Chen et al.'s work 2018).

By consolidating all these theoretical cues, we further give principled expressions for the gate function. One may expect to endow the gates some learning capacities, thus we let

$$g_t(x^{(t)}, y|\Lambda_g^{(t)}) = 1 + \kappa_t(x^{(t)}, y|\Lambda_g^{(t)}) = 1 + \mu_t b^{(t-1)} f_t(x^{(t)}), \tag{15}$$

in which $\mu_t \in \mathbb{R}$ is a parameter to be learned, $b^{(t-1)}$ is threshold of the $(t-1)$-th layer, and $f_t(x^{(t)})$ is a newly introduced function constrained not to be greater than $1/|x^{(t)}|$. We are going to evaluate different choices for the function $f_t(x^{(t)})$ in the supplementary material, e.g.,

$$\begin{aligned} \text{the piece-wise linear function: } & f_t(x^{(t)}) = \text{ReLU}(1 - \text{ReLU}(\nu_t |x^{(t)}|)), \\ \text{the inverse proportional function: } & f_t(x^{(t)}) = 1/(\nu_t |x^{(t)}| + \epsilon), \\ \text{the exponential function: } & f_t(x^{(t)}) = \exp(-\nu_t |x^{(t)}|), \end{aligned} \tag{16}$$

in which $\nu_t \in \mathbb{R}$ is a parameter to be learned, and ϵ is a tiny positive scalar introduced to avoid zero being divided. All the learnable parameters are thus collected as $\Lambda_g^{(t)} = \{\mu_t, \nu_t\}$.

3.1.1 NO FALSE POSITIVE?

Our previous theoretical results show that the performance of LISTA can be improved by using a gain gate, as long as the gate function satisfies conditions in Eq. (12) and (13), and no "false positive" is encountered. However, it is not always true in practice. Our experimental results also show that when the inverse proportional function is adopted as gain gates in lower layer for LISTA, the performance of our gated LISTA may even degrade. We conjecture that such contradiction to the theoretical results may be owing to impractical assumptions. In this subsection, we try to relax the assumption about no "false positive", and we further found that a tighter bound can be achieved with a more reasonable assumption instead. Through theoretical analysis as follows, we show that the inverse proportional gain function should better be only adopted in higher layers. For clarity of the result, we would like to introduce the following definition first.

Definition 2. *Given a model with Θ, in which $b^{(t)} = \Gamma\mu(A) \sup_{x_s} \|x^{(t)} \odot g_t(x^{(t)}, y|\Lambda_g^{(t)}) - x_s\|_1$, we introduce $\omega_{t+1}(k|\Theta)$ to characterize its relationship with the false positive rate, which is*

$$\omega_{t+1}(k_{t+1}|\Theta) = \sup_{\forall x_s, |supp(\tilde{x}^{(t+1)}) \cup supp(x_s)| \leq |supp(x_s)| + k_{t+1}} \Gamma,$$

in which $\tilde{x}^{(t+1)} := s_{b^{(t)}}(W^{(t)}(x^{(t)} \odot g_t(x^{(t)}, y|\Lambda_g^{(t)}) - x_s))$, and $k_{t+1} \geq 0$ is the desired maximal number of "false positive" of $x^{(t+1)}$.

The above definition applies to both the standard LISTA and LISTA with gain gates (we can let the gate function be an identity function to achieve a standard LISTA). We first analyze the convergence of LISTA without gates. We present theoretical results as follows on the basis of similar assumptions (including Assumption 1, $U^{(t)} \in \mathcal{W}(A)$, and $W^{(t)} = I - U^{(t)}A$), but with a different requirement for $b^{(t)}$ from Proposition 1.

Theorem 3. *If $b^{(t)} = \omega_{t+1}(k_{t+1}|\Theta)\mu(A) \sup_{x_s} \|x^{(t)} - x_s\|_1$ is achieved, and $\exists 0 < k_0^{(t)} < s$ such that $\omega_t(k_0^{(t)}|\Theta) < 1 - 1/(s - k_0^{(t)})$, then there exists "false positive" with $0 < k_t < s$ and*

$$\|x^{(t)} - x_s\|_2 \leq sB \exp(\sum_{i=1}^{t} c_i^*),$$

in which $c_i^* < \log((2s-1)\mu(A))$.

It can be seen that when we relax the assumption about no "false positive" and further reduce the value of the threshold $b^{(t)}$, the error bound of LISTA becomes even lower. Obviously, the previous bound of LISTA with gain gates in Theorem 2 is not necessarily lower than the tighter bound of a standard LISTA in Theorem 3, which well explains the contradiction of theoretical and empirical results. Here we re-deduce the error bound of our gated LISTA with the inverse proportional function in the following theorem. Note that we still have $U^{(t)} \in \mathcal{W}(A)$, $W^{(t)} = I - U^{(t)}A$ and Assumption 1.

Theorem 4. *Suppoes that* $\min_{i \in supp(x_s)} |(x_s)_i| \geq \sigma > 0$, *if* $b^{(t)} = \omega_{t+1}(k_{t+1}|\Theta)\mu(A) \sup_{x_s} \|x^{(t)} \odot g_t(x^{(t)}, y|\Lambda_g^{(t)}) - x_s\|_1$ *is achieved and* $\exists 0 < k_0^{(t)} < s$ *such that* $\omega_t(k_0^{(t)}|\Theta) < 1 - 1/(s - k_0^{(t)})$, *then*

$$\|x^{(t)} - x_s\|_2 \leq sB \exp(\sum_{i=1}^{t-1} c'_i + c'^{*}_t),$$

in which $c'^{*}_t < \log((2s-1)\mu(A))$. $\exists t_0 = \lceil \log(\frac{sB}{\sigma}) / \log(\frac{1}{(2s-1)\mu(A)}) \rceil$ *if the scaling factor* μ_i *of the gate have* $\mu_i = 0$ *for* $i < t_0$, $0 < k_i < s$, *then* $c'_i = c^*_i$, *and if* $1 - \omega_i(s|\Theta) < \mu_i \leq 1$ *for* $i \geq t_0$, $k_i = 0$, *then* $c'_i < c^*_i$.

We can conclude from Theorem 4 that, a) a gain gate expressed by the inverse proportional function should be applied to deeper layers in LISTA, rather than lower layers, b) when using the function, there indeed exists no "false positive" (i.e., $k_i = 0$) in deeper layer. We follow such guidelines in the implementation of our gated LISTA. In addition, we observe that unlike the inverse proportional function, other considered functions show consistent performance gains on both lower and higher layers, hence we attempt to utilize them on lower layers in combination with the inverse proportional function powered gain gates on the other layers. In the experiments, we choose the ReLU-based piece-wise linear function, and it is uniformly applied to the first 10 layers. Apparently, there are different combinations for the gain gate functions. We will experimentally compare their performance in the Appendix.

3.2 Sparse Coding with Overshoot Gates

Unlike the gain gates that are incorporated before performing the estimation at each step, the overshoot gates act more like on the output, which can be viewed as learnable boosts to the estimations:

$$\begin{aligned} \tilde{x}^{(t+1)} &= s_{b^{(t)}}(W^{(t)}x^{(t)} + U^{(t)}y), \\ x^{(t+1)} &= o_t(x^{(t)}, y|\Lambda_o^{(t)}) \odot \tilde{x}^{(t+1)} + (1 - o_t(x^{(t)}, y|\Lambda_o^{(t)})) \odot x^{(t)}. \end{aligned} \tag{17}$$

The gate function $o_t(\cdot, \cdot|\Lambda_o^{(t)}) : \{\mathbb{R}^n, \mathbb{R}^m\} \to \mathbb{R}^n$ outputs an n-dimensional vector and $\Lambda_o^{(t)}$ collects all the trainable parameters in the function, akin to a dedicated update GRU gate (Cho et al., 2014).

Our motivation comes from analyses of ISTA, whose update can be viewed as $x^{(t)} + \eta(x^{(t+1)} - x^{(t)})$, in which $\eta = 1$ is a constant step size. We argue that $\eta = 1$ may not be the most suitable choice and the following theorem makes this formal. We have the proposition which analyzes the update rule of ISTA and $\eta^* := \arg\min_\eta f(\eta(x^{(t+1)} - x^{(t)}) + x^{(t)}, y) + \lambda \|\eta(x^{(t+1)} - x^{(t)}) + x^{(t)}\|_1$.

Proposition 2. *(Requirement of overshoots) For* $\min_x f(x, y) + \lambda\|x\|_1$, *in which* $f(x, y)$ *is convex with respect to* x *and* $\gamma I - \nabla_x^2 f(x) \succ 0$ *holds for all* x, *if the update rule in Eq. (4) is adopted, then we have* $\eta^* \geq 1$. *In addition, if supp*$(x^{(t)}) \subset$ *supp*$(x^{(t+1)})$, *then we further have* $\eta^* > 1$.

See also Figure 3 for an illustration for the issue with $\eta = 1$ as concerned. Since the optimization procedure of ISTA inspires the network architecture in LISTA, the theoretical result in Proposition 2 that requires a boost in η for superior performance also inspires us to design specific overshoot gates for LISTA. Having noticed that an essential principle we have obtained is to let $\eta \geq 1$ (or $\eta > 1$), we may expect the output of the gate function to be greater than or at least equal to 1. To achieve the goal, we can try different expressions for it, e.g.,

$$o_t(x^{(t)}, y|\Lambda_o^{(t)}) = 1 + a_o\sigma(W_o x^{(t)} + U_o y) \left| \sum_i y_i \right| \tag{18}$$

with $\Lambda_o^{(t)} = \{a_o, W_o, U_o\}$ and $\sigma(\cdot)$ being the sigmoid function. The principle of our overshoot gate is similar to that of some momentum-based methods, e.g., FISTA (Beck & Teboulle, 2009) and

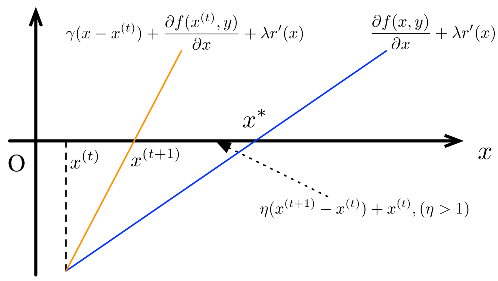


Figure 3: The derivative function (illustrated in blue) of $f(x, y) + \lambda r(x)$, in which $r(x) = \|x\|_1$, is monotonic owing to the convexity of $f(x, y)$ and $r(x)$, and its output should be consistently smaller than the derivative (illustrated in orange) of the upper bound in absolute value. Let x^* be the optimal solution to the problem, then we know from the figure that the estimation with a standard ISTA update (i.e., $\eta = 1$) normally "lags behind".

LFISTA (Moreau & Bruna). However, the fundamental difference between these methods and ours is that, (L)FISTA considers that the momentum term should be independent of the current input, i.e., being time invariant, while the output of the overshoot gate is a function of the t-th estimation and y, hence being time-varying. The design of our overshoot gate endows the network higher capacity to learn. Experimental results in the Appendix confirm the superiority of our method.

4 Experiments

In this section, we perform experiments to confirm our theoretical results and evaluate the performance of our gated sparse coding networks. The validation of our theoretical results are performed on synthetic data. We set $m = 250$, $n = 500$, and we sample the elements of the dictionary matrix A randomly from a standard Gaussian distribution. The position of non-zero elements of the sparse vector x_s is determined by a Bernoulli sampling with a probability of 0.1 (which means nearly 90% of the elements are set to be zero). Different noise levels and condition numbers are considered in the sparse coding experiments, and we synthesize 1000 samples to constitute the test set for each combination of them. Our training settings mostly follows those of Chen et al.'s (2018). For the proposed gated LISTA, we set $d = 16$ and let the parameters $\{b^{(t)}\}$ not be shared between different layers under all circumstances. We are going to compare with strong LISTA baselines. For all deep learning-based methods, the parameter matrices $\{W^{(t)}, U^{(t)}\}$ are not shared between different layers and the coupled constraints $W^{(t)} = I - U^{(t)}A$, $\forall t$ are satisfied.

Our evaluation metric for sparse coding is the normalized MSE (NMSE) (Chen et al., 2018):

$$\text{NMSE}(x, x_s) = 10\log_{10}(\|x - x_s\|_2^2 / \|x_s\|_2^2). \tag{19}$$

4.1 Simulation Experiments

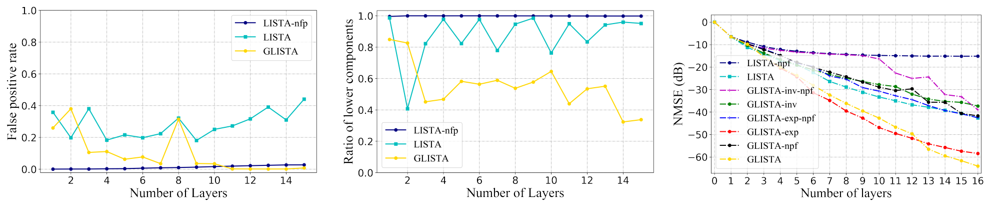
4.1.1 Validation of Theoretical Results

Validation of Proposition 1: We first confirm Proposition 1. In order to ensure that LISTA fulfills the assumption about no "false positive", we introduce an auxiliary loss into the learning object as:

$$\lambda \sum_t \sum_{j \notin \text{supp}(x_s)} |(x^{(t)})_j|. \tag{20}$$

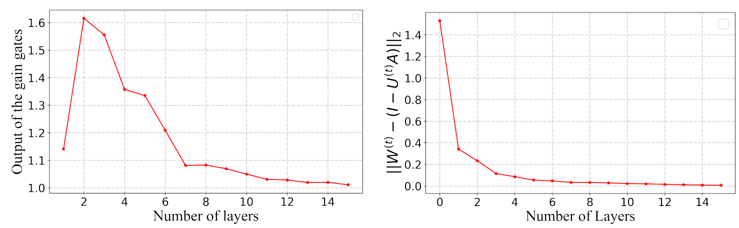
We formally introduce the false positive rate (FPR) as FPR $= \frac{|\text{supp}(x^{(t)}) \cup \text{supp}(x_s)| - |\text{supp}(x_s)|}{|\text{supp}(x^{(t)})|}$ and try to approach no "false positive" (i.e., LISTA-nfp) by setting $\lambda = 5.0$ in the experiment. [1] Check Figure 4 for an illustrative comparison between different models, we see LISTA-nfp achieves almost no "false positive" in practice in Figure 4(a), but its convergence is slower as demonstrated in Figure 4(c), which is consistent with our result in **Theorem 3**. In addition, we also see in Figure 4(b)

[1] The FPR here is slightly different from the general false positive rate by calculating only in the obtained positive code components.



(a) FPR vs # of layers (b) Ratio of lower components (c) NMSE vs # of layers

Figure 4: Experimental results confirming our Proposition 1 and Theorem 2.



(a) (b)

Figure 5: Experimental results confirming our Theorem 1. It can be observed that: (a) the gate output converges to 1, and (b) LISTA with our gain gates converges as expected.

that without "false positive", the code components in LISTA estimations are almost always less than those of the ground-truth, which confirms our Proposition 1.

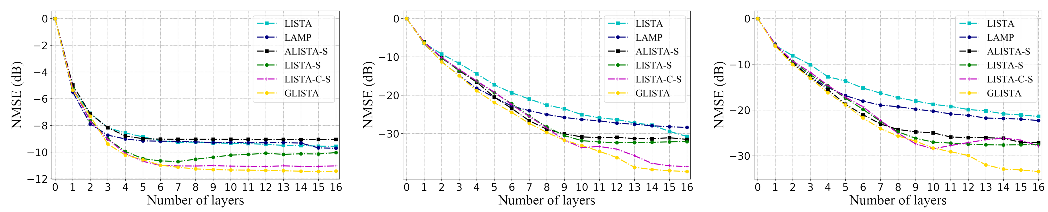
Validation of Theorem 1: We aim to calculate $\|W^{(t)}D - (I - U^{(t)}A)\|_2$ using a gated LISTA with the introduced ReLU-based piece-wise linear gain gate function [2]. To accomplish this task, we need to first evaluate the output of our gate function. In fact, as demonstrated in our proof, the bias term converges to zero when $t \to \infty$, thus we may expect the output of the gates to converge to 1. We also show such a trend in Figure 5(a). Consequently, the matrix D is supposed to be an identity matrix in the end and we can calculate $\|W^{(t)} - (I - U^{(t)}A)\|_2$ as a surrogate. In Figure 5(b), it indeed converges to zero in the end and the results confirm the theorem.

Validation of Theorem 2: We apply three kinds of gated LISTA with a combination of gain gate function (i.e., what has been introduced in Section 3.1.1), the exponential function, and the inverse proportional function respectively to verify our theoretical results. They were named as GLISTA (which is the abbreviation of gated LISTA), GLISTA-exp, GLISTA-inv, respectively. From Figure 4(c), we see that when the models with gain gates has no "false positive", all of them are superior to the standard LISTA without "false positive" as well, which is consistent with the conclusion of Theorem 2. In addition, from Figure 4(a), we can also see that there actually exists "false positive" in the lower layers of GLISTA*, but even without the auxiliary loss term, the FPR of our GLISTA and it variants approaches zero in the higher layers, which is in good agreement with the conclusion of **Theorem 4**.

4.1.2 Comparison with Competitors

Compared with other state-of-the-art methods: We consider four state-of-the-arts: LISTA with support selections (namely LISTA-C-S and LISTA-S, with and without the coupled constraint) (Chen et al., 2018), analytic LISTA with support selections (ALISTA-S) (Liu et al., 2018), and learned AMP (LAMP) (Borgerding et al., 2017) for comparison, and their official implementations are directly used. The hyper-parameters are set following the papers (Borgerding et al., 2017; Chen et al., 2018). We compare our GLISTA with these competitive methods under different levels of noises (including the signal-to-noise ratios (SNRs) being equal to 40dB, 20dB, and 10dB) and different condition numbers (including 3, 30, and 100, with SNR=40dB). See Figure 6 for comparisons between LISTA, LAMP, LISTA-S, LISTA-C-S, ALISTA-S, and our GLISTA under some of the settings. Obviously, the introduced gates facilitate LISTA significantly, and the concerned NMSE diminishes the fastest using GLISTA. See our Appendix for comparisons of final perfor-

[2] Other functions can be adopted and the same results can be obtained.



(a) SNR=10dB (b) condition number=30 (c) condition number=100

Figure 6: Comparison of sparse coding methods under different settings. Our GLISTA consistently outperforms the competitors in almost all test cases with different numbers of layers.

mance after multiple runs and the results in other settings (i.e., SNR: 20dB, 40dB, and condition number: 3). We know from these results that using the gain gates solely can already outperforms existing state-of-the-arts, while incorporating the overshoot gates additionally may further boost the performance, as testified in the Appendix.

Applying our method to variants of LISTA: We also try adopting the introduced gates into some variants of LISTA to verify their "generalization ability". Specifically, we incorporate the overshoot gates to LFISTA (Moreau & Bruna) and ALISA (Liu et al., 2018) to obtain GFLISTA and AGLISTA, respectively. Since ALISTA is suggested to be implemented with support set selection in the paper, i.e. ALISTA-S, we also compare with it. The experiment is performed under different levels of noises (40dB, 20dB, 10dB). As can be seen from Table 1, the performance of models with our gates is significantly better, which verifies that our method generalizes well. The same result can be obtained using our gain gates.

Table 1: Comparison of LISTA and its variants (with and without gates) under different noise levels.

SNR	LISTA	GLISTA	LFISTA	GLFISTA	ALISTA	AGLISTA	ALISTA-S
40	-38.72	**-45.22**	-37.84	**-38.30**	-37.86	**-42.30**	-41.86
20	-18.65	**-23.08**	-20.90	**-22.00**	-17.38	**-20.13**	-20.00
10	-9.42	**-11.41**	-10.67	**-11.20**	-8.39	**-9.13**	-9.04

4.2 Photometric Stereo Analysis

We now test on a more practical task, i.e., photometric stereo analysis, using sparse coding. For a 3D object with Lambertian surface, if there are q different light conditions, a camera or some other kinds of sensors can obtain q different observations, all with noises caused by shadows and specularities. The observations can be represented as a vector $o \in \mathbb{R}^q$ for estimating the norm vector $n \in \mathbb{R}^3$ at any position on the surface. In general, it is formulated as $o = \rho L n + e$, in which $L \in \mathbb{R}^{3\times q}$ represents the normalized light directions, $e \in \mathbb{R}^q$ is a noise which is often sparse, $\rho \in \mathbb{R}$ representes the albedo reflectivity. Our task is to obtain n from o. More detailed descrptions of the task can be found in Xin et al.'s paper (2016). We mostly follow the settings in Xin et al.'s paper, except that we test with $q = 15, 25, 35$, and let 40% of the elements of e be zero. We use GLISTA here to estimate e and the final result for n is calculated as $L^\dagger(o - e)$. Our method is compared with LISTA and two traditional methods, i.e. the original least square (LS) and least L1, in Table 2. Our evaluation metric is the mean error in degree and it is calculated using the bunny picture (Xin et al., 2016).

Table 2: Mean error in degree with different number of observations ($q = 15, 25, 35$).

q	LS	L1	LISTA	GLISTA
35	5.37	1.39	0.0266	**0.00198**
25	5.60	2.03	0.0313	**0.00498**
15	6.09	4.25	0.384	**0.0359**

5 Conclusion

In this paper, we study LISTA for solving sparse coding problems. We discover its potential weaknesses and introduce gated mechanisms to address them accordingly. In particular, we theoretically prove that LISTA with gain gates can achieve faster convergence than the standard LISTA. We also discover that LISTA (with or without gates) can obtain lower reconstruction errors under a weaker assumption of "false positive" in its code estimations. It helps us improve the convergence analyses to achieve more solid theoretical results, which have been perfectly confirmed in simulation experiments. The effectiveness of our introduced gates is verified in a variety of sparse coding experiments and the state-of-the-art performance is achieved. In the future, we aim to extend the method to convolutional neural networks to deal with more complex tasks.

REFERENCES

Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009.

Mark Borgerding, Philip Schniter, and Sundeep Rangan. Amp-inspired deep networks for sparse linear inverse problems. *IEEE Transactions on Signal Processing*, 65(16):4293–4308, 2017.

Xiaohan Chen, Jialin Liu, Zhangyang Wang, and Wotao Yin. Theoretical linear convergence of unfolded ista and its practical weights and thresholds. In *Advances in Neural Information Processing Systems*, pp. 9061–9071, 2018.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *EMNLP*, 2014.

Ingrid Daubechies, Michel Defrise, and Christine De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 57(11):1413–1457, 2004.

Bradley Efron, Trevor Hastie, Iain Johnstone, Robert Tibshirani, et al. Least angle regression. *The Annals of statistics*, 32(2):407–499, 2004.

Michael Elad and Michal Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image processing*, 15(12):3736–3745, 2006.

Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448, 2015.

Raja Giryes, Yonina C Eldar, Alex M Bronstein, and Guillermo Sapiro. Tradeoffs between convergence speed and reconstruction accuracy in inverse problems. *IEEE Transactions on Signal Processing*, 66(7):1676–1690, 2018.

Karol Gregor and Yann LeCun. Learning fast approximations of sparse coding. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, pp. 399–406. Omnipress, 2010.

Hao He, Bo Xin, Satoshi Ikehata, and David Wipf. From bayesian sparsity to gated recurrent nets. In *Advances in Neural Information Processing Systems*, pp. 5554–5564, 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

John R Hershey, Jonathan Le Roux, and Felix Weninger. Deep unfolding: Model-based inspiration of novel deep architectures. *arXiv preprint arXiv:1409.2574*, 2014.

Geoffrey Hinton, Li Deng, Dong Yu, George Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Brian Kingsbury, et al. Deep neural networks for acoustic modeling in speech recognition. *IEEE Signal processing magazine*, 29, 2012.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015.

Jialin Liu, Xiaohan Chen, Zhangyang Wang, and Wotao Yin. Alista: Analytic weights are as good as learned weights in lista. 2018.

Thomas Moreau and Joan Bruna. Understanding trainable sparse coding via factorization. In *In International Conference on Learning Representation*.

Jeremias Sulam, Aviad Aberdam, Amir Beck, and Michael Elad. On multi-layer basis pursuit, efficient algorithms and convolutional neural networks. *IEEE transactions on pattern analysis and machine intelligence*, 2019.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

Zhangyang Wang, Qing Ling, and Thomas S Huang. Learning deep l0 encoders. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

Bo Xin, Yizhou Wang, Wen Gao, David Wipf, and Baoyuan Wang. Maximal sparsity with deep networks? In *Advances in Neural Information Processing Systems*, pp. 4340–4348, 2016.

Jianchao Yang, John Wright, Thomas S Huang, and Yi Ma. Image super-resolution via sparse representation. *IEEE transactions on image processing*, 19(11):2861–2873, 2010.

Jian Zhang and Bernard Ghanem. Ista-net: Interpretable optimization-inspired deep network for image compressive sensing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1828–1837, 2018.

APPENDIX

A PROOF OF THEOREMS AND PROPOSITIONS

Before we delve deeply into the proof, we first give some importance notations. We define $\mathcal{S}$ as the support of the vector x_s, i.e. $\mathcal{S} = \text{supp}(x_s)$, and let $|\mathcal{S}|$ denote the number of elements in the set $\mathcal{S}$. For a vector that shares the same size with x_s, say z, we denote by $z_{\mathcal{S}} \in \mathbb{R}^{|\mathcal{S}|}$ a vector that keeps the elements with indices of z in $\mathcal{S}$ and removes the others. If the vectors have been introduced with subscripts already, e.g. x_s, we use $(x_s)_{\mathcal{S}}$ to denote vectors obtained in such a manner. For a square matrix with the same number of row and column as the size of x_s, say M, $M(\mathcal{S}, \mathcal{S})$ is its principal minor with the index set formed by removing rows and columns whose indices are in $\mathcal{S}$. Assume a vector x with no zero elements, sign$(\cdot)$ is defined as $(\text{sign}(x))_i = x_i/|x_i|$, i.e. $(\text{sign}(x))_i = 1$ when $x_i > 0$, and $(\text{sign}(x))_i = -1$ when $x_i < 0$.

A.1 PROOF OF PROPOSITION 1

Recall that the update rule of LISTA is $x^{(0)} = \mathbf{0}$ and

$$x^{(t+1)} = s_{b^{(t)}}(W^{(t)}x^{(t)} + U^{(t)}y), \quad t = 0, \cdots, d-1. \tag{21}$$

Proof. Recall the definition of $\mathcal{S}$ is $\mathcal{S} = \text{supp}(x_s)$. For the shrinking function $s_{b^{(t)}}(x) = \text{sign}(x)(|x| - b^{(t)})_+ = x - bh(x)$, where $h(x) = 1$ if $x > 0$, $h(x) = -1$ if $x < 0$, and $h(x) \in [-1, 1]$ if $x = 0$.

We use Mathematical Induction to prove $\text{supp}(x^{(t)}) \subset \mathcal{S}, \forall t = 0, 1, \ldots, d-1$. We assume $\text{supp}(x^{(t)}) \subset \mathcal{S}$. From the calculation of $x_i^{(t+1)}$, as $W^{(t)} = I - U^{(t)}A$ there is

$$\begin{aligned} x_i^{(t+1)} &= s_{b^{(t)}}((W^{(t)}x_t + U^{(t)}y)_i) \\ &= s_{b^{(t)}}((W^{(t)}x_t + U^{(t)}Ax_s)_i) \\ &= s_{b^{(t)}}(((I - U^{(t)}A)(x^{(t)} - x_s))_i + (x_s)_i) \\ &= ((I - U^{(t)}A)(x^{(t)} - x_s))_i + (x_s)_i - b^{(t)}h(x_i^{(t+1)}). \end{aligned} \tag{22}$$

When the $i \notin \mathcal{S}$, $(x_s)_i = 0$. Let's assume $x_i^{(t+1)} \neq 0$, then $h(x_i^{(t+1)}) = \text{sign}(x_i^{(t+1)})$. Multiply the two sides of the Eq. (22) by $\text{sign}(x_i^{(t+1)})$, as the $b^{(t)} = \mu(A)\sup_{x_s} \|x^{(t)} - x_s\|_1$, there will be

$$\begin{aligned} |x_i^{(t+1)}| &= |((I - U^{(t)}A)(x^{(t)} - x_s))_i \text{sign}(x_i^{(t+1)}) - b^{(t)}| \\ &= ((I - U^{(t)}A)(x^{(t)} - x_s))_i \text{sign}(x_i^{(t+1)}) - \mu(A)\sup_{x_s} \|x^{(t)} - x_s\|_1 \\ &\leq \mu(A)\|x^{(t)} - x_s\|_1 - \mu(A)\sup_{x_s} \|x^{(t)} - x_s\|_1 \leq 0, \end{aligned} \tag{23}$$

which is in conflict with $x_i^{(t+1)} \neq 0$. Therefore, the $x_i^{(t+1)} = 0$, when $i \notin \mathcal{S}$, i.e., $\text{supp}(x^{(t+1)}) \subset \mathcal{S}$. As $x^{(0)} = \mathbf{0} \subset \mathcal{S}$, the $\text{supp}(x^{(t)}) \subset \mathcal{S}, \forall t$. The no "false positive" property has been proved.

According to Eq. (22), as support set of x_s and $x^{(t)}$ are the subsets of $\mathcal{S}$, there is

$$\begin{aligned} x_i^{(t+1)} - (x_s)_i &= ((I - U^{(t)}A)(x^{(t)} - x_s))_i - b^{(t)} h(x_i^{(t+1)}) \\ &= \sum_{j \in \mathcal{S}} (I - U^{(t)}A)_{ij} (x_j^{(t)} - (x_s)_j) - b^{(t)} h(x_i^{(t+1)}) \\ |x_i^{(t+1)} - (x_s)_i| &\leq |\sum_{j \in \mathcal{S}} (I - U^{(t)}A)_{ij} (x_j^{(t)} - (x_s)_j)| + b^{(t)}. \end{aligned} \tag{24}$$

As $\text{supp}(x^{(t+1)}) \subset \mathcal{S}$, accumulate all $|x_i^{(t+1)} - (x_s)_i|$ $(i \in \mathcal{S})$ in Eq. (24), there is

$$\begin{aligned} \|x^{(t+1)} - x_s\|_1 &\leq \sum_{i \in \mathcal{S}} \sum_{j \in \mathcal{S}} (I - U^{(t)}A)_{ij} (x_j^{(t)} - (x_s)_j) + |\mathcal{S}| b^{(t)} \\ &\leq \sum_{i \in \mathcal{S}} \sum_{j \in \mathcal{S}, i \neq j} |(I - U^{(t)}A)_{ij}| |x_j^{(t)} - (x_s)_j| + |\mathcal{S}| b^{(t)} \\ &\leq (|\mathcal{S}| - 1) \mu(A) \|x^{(t)} - x_s\|_1 + |\mathcal{S}| b^{(t)} \end{aligned} \tag{25}$$

The second equation is because of $U^{(t)} \in \mathcal{W}(A)$, so that $|W_{i,:} A_{:,j}| \leq \mu(A)$ when $i \neq j$ and $|W_{i,:} A_{:,j}| = 1$ when $i = j$. Substitute $b^{(t)} = \mu(A) \sup_{x_s} \|x^{(t)} - x_s\|_1$ into Eq. (25), and take the supremum of Eq. (25), there is

$$\begin{aligned} \sup_{x_s} \|x^{(t+1)} - x_s\|_1 &\leq (|\mathcal{S}| - 1) \mu(A) \sup_{x_s} \|x^{(t)} - x_s\|_1 + |\mathcal{S}| \mu(A) \sup_{x_s} \|x^{(t)} - x_s\|_1 \\ &\leq (2|\mathcal{S}| - 1) \mu(A) \sup_{x_s} \|x^{(t)} - x_s\|_1 \\ &\leq ((2|\mathcal{S}| - 1) \mu(A))^{t+1} \sup_{x_s} \|x^{(0)} - x_s\|_1. \end{aligned} \tag{26}$$

Let $c = \log((2|\mathcal{S}| - 1)\mu(A))$, the l_2 error bound of t-th layer in LISTA should be calculated as

$$\begin{aligned} \|x^{(t)} - x_s\|_2 \leq \|x^{(t)} - x_s\|_1 &\leq \sup_{x_s} \|x^{(t)} - x_s\|_1 \\ &\leq ((2|\mathcal{S}| - 1)\mu(A))^t \sup_{x_s} \|x^{(0)} - x_s\|_1 \\ &= \exp(ct) \sup_{x_s} \|x^{(0)} - x_s\|_1 \\ &\leq sB \exp(ct), \end{aligned} \tag{27}$$

where the last equation is deduced for $(x_s)_i \leq B$, and $\|x_s\|_0 \leq s$. The linear convergence has been proved.

Refer to the Eq. (24), we concentrate on $i \in \mathcal{S}$

$$x_i^{(t+1)} - (x_s)_i = (I - U^{(t)}A)(x^{(t)} - x_s)_i - b^{(t)} h(x_i^{(t+1)}). \tag{28}$$

If $x_i^{(t+1)} = 0$, there must be $|x_i^{(t+1)}| = 0 \leq (x_s)_i$, and $x_i^{(t+1)}(x_s)_i = 0$.

If $x_i^{(t+1)} > 0$, the according to Eq. (28), $x_i^{(t+1)} - (x_s)_i = (I - U^{(t)}A)(x^{(t)} - x_s)_i - b^{(t)} = (I - U^{(t)}A)(x^{(t)} - x_s)_i - \sup_{x_s} \|x^{(t)} - x_s\|_1 \leq 0$, i.e. ,$0 < x_i^{(t+1)} \leq (x_s)_i$, $|x_i^{(t+1)}| \leq |(x_s)_i|$, and $x_i^{(t+1)}(x_s)_i > 0$.

If $x_i^{(t+1)} < 0$, the according to Eq. (28), $x_i^{(t+1)} - (x_s)_i = (I - U^{(t)}A)(x^{(t)} - x_s)_i + b^{(t)} = (I - U^{(t)}A)(x^{(t)} - x_s)_i + \sup_{x_s} \|x^{(t)} - x_s\|_1 \geq 0$, i.e. ,$0 > x_i^{(t+1)} \geq (x_s)_i$, $|x_i^{(t+1)}| \leq |(x_s)_i|$, and $x_i^{(t+1)}(x_s)_i > 0$.

In conclusion, we can obtain $|x_i^{(t+1)}| \leq |(x_s)_i|$ for all the situations.

□

A.2 Proof of Theorem 1

Recall that the update rule of LISTA with gain gates is $x^{(0)} = \mathbf{0}$ and

$$x^{(t+1)} = s_{b^{(t)}}(W^{(t)}(g_t(x^{(t)}, y|\Lambda_g^{(t)}) \odot x^{(t)}) + U^{(t)}y). \tag{29}$$

Proof. According to definition of the shrinking function $s_{b^{(t)}}(\cdot)$ and $y = Ax_s$, Eq. (10) is

$$\begin{aligned} x^{(t+1)} &= s_{b^{(t)}}(W^{(t)}(g_t(x^{(t)}, y|\Lambda_g^{(t)}) \odot x^{(t)}) + U^{(t)}y) \\ &= W^{(t)}g_t(x^{(t)}, y|\Lambda_g^{(t)}) \odot x^{(t)} + U^{(t)}y - b^{(t)} \odot h(x^{(t+1)}) \\ &= W^{(t)}\text{diag}(g_t(x^{(t)}, y|\Lambda_g^{(t)}))x^{(t)} + U^{(t)}Ax_s - b^{(t)} \odot h(x^{(t+1)}) \\ &= W^{(t)}\text{diag}(g_t(x^{(t)}, y|\Lambda_g^{(t)}))x^{(t)} + U^{(t)}Ax_s - b^{(t)} \odot h(x^{(t+1)}). \end{aligned} \tag{30}$$

Define $g_t(x_s) = (g_t(x_s, y|\Lambda_g^{(t)}))$ when $t \to \infty$. In the main part of the Theorem 1, $\forall x_s$ satisfying $\|x_s\|_0 \le s$ is the fixed point of Eq.(10) when $t \to \infty$. Eq. (30) is

$$x_s = W^{(t)}\text{diag}(g_\kappa(x_s))x_s + U^{(t)}Ax_s - b^{(t)} \odot h(x_s). \tag{31}$$

The equation group of the indices in $\mathcal{S}$ in Eq. (31) is

$$\begin{aligned} (x_s)_\mathcal{S} &= (((W^{(t)}\text{diag}(g_\kappa(x_s)) + U^{(t)}A)x_s)_\mathcal{S} - b_\mathcal{S}^{(t)} \odot h((x_s)_\mathcal{S}) \\ &= (W^{(t)}(\mathcal{S},\mathcal{S})\text{diag}(g_\kappa((x_s)_\mathcal{S})) + (U^{(t)}A)(\mathcal{S},\mathcal{S}))(x_s)_\mathcal{S} - b_\mathcal{S}^{(t)} \odot h((x_s)_\mathcal{S}). \end{aligned} \tag{32}$$

Let $(x_s)_\mathcal{S} \to 0$ but $(x_s)_\mathcal{S} \neq 0$ so that $h((x_s)_\mathcal{S}) = \text{sign}((x_s)_\mathcal{S})$. As $W^{(t)}$, $U^{(t)}$, A and $g_\kappa(x_s) = g_t(x_s, y|\Lambda_g^{(t)})$ are bounded, the right hand side of Eq. (32) is also tend to 0, which is

$$b_\mathcal{S}^{(t)} \to 0, \quad \text{as} \quad t \to \infty. \tag{33}$$

As the $\mathcal{S}$ can be selected arbitrarily as long as $|\mathcal{S}| \le s$, $b^{(t)}$ also satisfies

$$b^{(t)} \to 0, \quad \text{as} \quad t \to \infty. \tag{34}$$

Substitute the $b_\mathcal{S}^{(t)}$ of Eq. (33) into Eq. (32), $(x_s)_\mathcal{S}$ is

$$(x_s)_\mathcal{S} = (W^{(t)}(\mathcal{S},\mathcal{S})\text{diag}(g_\kappa((x_s)_\mathcal{S}))_\mathcal{S} + (U^{(t)}A)(\mathcal{S},\mathcal{S}))(x_s)_\mathcal{S}, \tag{35}$$

where the $W^{(t)}(\mathcal{S},\mathcal{S})$ is defined at start of this section. Eq. (35) is

$$\begin{aligned} (I - U^{(t)}A)(\mathcal{S},\mathcal{S})(x_s)_\mathcal{S} &= W^{(t)}(\mathcal{S},\mathcal{S})\text{diag}(g_\kappa((x_s)_\mathcal{S}))(x_s)_\mathcal{S}, \\ (I - U^{(t)}A)(\mathcal{S},\mathcal{S})(x_s)_\mathcal{S} &= W^{(t)}(\mathcal{S},\mathcal{S})\text{diag}((x_s)_\mathcal{S})g_\kappa((x_s)_\mathcal{S}), \\ \text{diag}(((x_s)_\mathcal{S})^{-1})(W^{(t)}(\mathcal{S},\mathcal{S}))^{-1}(I - U^{(t)}A)(\mathcal{S},\mathcal{S})(x_s)_\mathcal{S} &= g_\kappa((x_s)_\mathcal{S}), \\ \text{diag}(((x_s)_\mathcal{S})^{-1})M(x_s)_\mathcal{S} &= g_\kappa((x_s)_\mathcal{S}), \end{aligned} \tag{36}$$

where $M = (W^{(t)}(\mathcal{S},\mathcal{S}))^{-1}(I - U^{(t)}A)(\mathcal{S},\mathcal{S})$. The i-th row and j-th column element in M is denoted as m_{ij}. From Eq. (36), $(g_\kappa(x_s))_\mathcal{S}$ is

$$\begin{aligned} (g_\kappa(x_s))_\mathcal{S} &= \begin{bmatrix} ((x_s)_\mathcal{S})_1^{-1} & & & \\ & ((x_s)_\mathcal{S})_2^{-1} & & \\ & & \dots & \\ & & & ((x_s)_\mathcal{S})_{|\mathcal{S}|}^{-1} \end{bmatrix} \begin{bmatrix} m_{11} & m_{12} & \dots & m_{1|\mathcal{S}|} \\ m_{21} & m_{22} & \dots & m_{2|\mathcal{S}|} \\ \dots & \dots & \dots & \dots \\ m_{|\mathcal{S}|1} & m_{|\mathcal{S}|2} & \dots & m_{|\mathcal{S}||\mathcal{S}|} \end{bmatrix} \\ &\quad \begin{bmatrix} ((x_s)_\mathcal{S})_1 \\ ((x_s)_\mathcal{S})_2 \\ \dots \\ ((x_s)_\mathcal{S})_{|\mathcal{S}|} \end{bmatrix} \\ &= \begin{bmatrix} \frac{\sum_{i=1}^{|\mathcal{S}|} m_{1i}((x_s)_\mathcal{S})_i}{((x_s)_\mathcal{S})_1} \\ \frac{\sum_{i=1}^{|\mathcal{S}|} m_{2i}((x_s)_\mathcal{S})_i}{((x_s)_\mathcal{S})_2} \\ \dots \\ \frac{\sum_{i=1}^{|\mathcal{S}|} m_{|\mathcal{S}|i}((x_s)_\mathcal{S})_i}{((x_s)_\mathcal{S})_{|\mathcal{S}|}} \end{bmatrix}. \end{aligned}$$

Assume $(x_s)_{\mathcal{S}} \to 0$, for $g_\kappa(x_s)$ is bounded, we can conclude that $m_{ij} = 0$, if $i \neq j$. From Eq. (37), the final form of $g_\kappa((x_s)_{\mathcal{S}})$ is formulated as

$$g_\kappa((x_s)_{\mathcal{S}}) = \begin{bmatrix} m_{11} \\ m_{22} \\ \cdots \\ m_{|\mathcal{S}||\mathcal{S}|} \end{bmatrix}. \tag{37}$$

From Eq. (37), we can conclude that $g_\kappa(x_s)_i$ is a constant if $i \in \mathcal{S}$, as the $\mathcal{S}$ could be arbitrary subset of $\{1, \cdots, n\}$ as long as $|\mathcal{S}| \leq s$. We could deduce that $g_\kappa(x_s)_i$ is constant $\forall i \in \{1, \ldots, n\}$ and $g_\kappa(x_s)$ must be constant vector, i.e.

$$\text{diag}(g_\kappa(x_s)) = D, \tag{38}$$

where D is an $n \times n$ constant matrix. The first part of conclusion of Theorem 1 has been proved.

Substitute $b^{(t)}$ in Eq. (33) and $\text{diag}(g_\kappa(x_s))$ in Eq. (38) into Eq. (31), Eq. (31) is.

$$\begin{aligned} x_s &= (W^{(t)}D + U^{(t)}A)x_s, \\ x_s &= Zx_s, \end{aligned} \tag{39}$$

where $Z = W^{(t)}D + U^{(t)}A = [Z_1, Z_2, \ldots, Z_n]$and the Z_i is the i-th column of Z.

Give a x_s satisfying only the i-th element of x_s is non-zero and all the other elements are equal to zero, i.e., $x_s = [0, 0, \ldots, \omega, \ldots, 0]^T = \omega e_i$, in which e_i is basis vector with only the i-th element being 1 and $\omega \neq 0$. Substitute the $x_s = \omega e_i$ into Eq. (39),

$$\begin{aligned} x_s &= Zx_s, \\ \omega e_i &= [Z_1, Z_2, \ldots, Z_n][0, 0, \ldots, \omega, \ldots, 0]^T, \\ \omega e_i &= \omega Z_i, \\ \omega(e_i - Z_i) &= \mathbf{0}. \end{aligned} \tag{40}$$

As the Eq. (40) should hold for $\forall \omega \neq 0$, we can deduce that $Z_i = e_i$. As the i is selected arbitrarily, $Z = W^{(t)}D + U^{(t)}A = [Z_1, Z_2, \ldots, Z_n] = [e_1, e_2, \ldots, e_n] = I$. Thus we have completed the proof and get

$$W^{(t)}D = (I - U^{(t)}A) \quad \text{as} \quad t \to \infty. \tag{41}$$

□

A.3 Proof of Theorem 2

Recall that the update rule of LISTA with gain gates is $x^{(0)} = \mathbf{0}$ and

$$x^{(t+1)} = s_{b^{(t)}}(W^{(t)}(g_t(x^{(t)}, y|\Lambda_g^{(t)}) \odot x^{(t)}) + U^{(t)}y). \tag{42}$$

Proof. We simplify the $g_t(x^{(t)}, y|\Lambda_g^{(t)})$ as $g_t(x^{(t)})$, and $\kappa_t(x^{(t)}, y|\Lambda_g^{(t)})$ as $\kappa_t(x^{(t)})$. According to the definition of gain gate in Eq.(42), we have

$$\begin{aligned} x^{(t+1)} &= s_{b^{(t)}}(W^{(t)}(x^{(t)} \odot g_t(x^{(t)}) + U^{(t)}y) \\ &= s_{b^{(t)}}(W^{(t)}(x^{(t)} \odot g_t(x^{(t)}) + U^{(t)}Ax_s) \\ &= s_{b^{(t)}}((I - U^{(t)}A)(x^{(t)} \odot g(x^{(t)}) - x_s) - x_s) \\ &= (I - U^{(t)}A)(x^{(t)} \odot g(x^{(t)}) - x_s) - x_s - b^{(t)}h(x^{(t+1)}). \end{aligned} \tag{43}$$

Simplify the $x^{(t)} \odot g(x^{(t)}) - x_s$ as $\Delta_g x^{(t)}$. For the i-th equation in Eq. (43), and $i \notin \mathcal{S}$, give the value of $b^{(t)} = \mu(A) \sup_{x_s} \|x^{(t)} \odot g(x^{(t)}) - x_s\|_1$, there is

$$\begin{aligned} x_i^{(t+1)} &= ((I - U^{(t)}A)(\Delta_g x^{(t)}))_i - b^{(t)}h(x_i^{(t+1)}) \\ &= ((I - U^{(t)}A)(\Delta_g x^{(t)}))_i - \mu(A) \sup_{x_s} \|\Delta_g x^{(t)}\|_1 h(x_i^{(t+1)}). \end{aligned} \tag{44}$$

With almost the same proof process in Theorem 1, we could deduce that

$$\mathrm{supp}(x^{(t+1)}) \subset x_s, \tag{45}$$

which is the no "false poistive" property.

Recall the Eq. (43) and substitute the $1 + \kappa_{t+1}(x^{(t+1)}) = g_{t+1}(x^{(t+1)})$:

$$\begin{aligned} x^{(t+1)} &= ((I - U^{(t)}A)(\Delta_g x^{(t)})) - b^{(t)}h(x^{(t+1)}) + x_s, \\ x^{(t+1)}(1 + \kappa_{t+1}(x^{(t+1)})) &= (I - U^{(t)}A)(\Delta_g x^{(t)}) - b^{(t)}h(x^{(t+1)}) + x_s + x^{(t+1)}\kappa_t(x^{(t+1)}) \\ \Delta_g x^{(t+1)} &= (I - U^{(t)}A)(\Delta_g x^{(t)}) - b^{(t)}h(x^{(t+1)}) + x^{(t+1)}\kappa_t(x^{(t+1)}). \end{aligned} \tag{46}$$

For $i \in \mathcal{S}$ but $i \notin \mathrm{supp}(x^{(t+1)})$, $x^{(t+1)} = 0$. Select the i-th equation in Eq. (46), there is

$$\begin{aligned} |\Delta_g x_i^{(t+1)}| &= ((I - U^{(t)}A)(\Delta_g x^{(t)}))_i - b^{(t)}h(x_i^{(t+1)}) \\ &\leq \mu(A) \sum_{j\in\mathcal{S}, j\neq i} |\Delta_g x_i^{(t)}| + |b^{(t)}|. \end{aligned} \tag{47}$$

For $i \in \mathcal{S}$ and $i \in \mathrm{supp}(x^{(t+1)})$, select the i-th equation in Eq. (46), there is

$$\begin{aligned} |\Delta_g x_i^{(t+1)}| &= ((I - U^{(t)}A)(\Delta_g x^{(t)}))_i - b^{(t)}h(x_i^{(t+1)}) + x^{(t+1)}\kappa_{t+1}(x^{(t+1)}) \\ &\leq \mu(A) \sum_{j\in\mathcal{S}, j\neq i} |\Delta_g x_i^{(t)}| - b^{(t)}\mathrm{sign}(x_i^{(t+1)}) + \mathrm{sign}(x_i^{(t+1)})(|x_i^{(t+1)}|\kappa_{t+1}(x^{(t+1)})) \\ &\leq \mu(A) \sum_{j\in\mathcal{S}, j\neq i} |\Delta_g x_i^{(t)}| + (|x_i^{(t+1)}|\kappa_{t+1}(x^{(t+1)}) - b^{(t)})\mathrm{sign}(x_i^{(t+1)}). \end{aligned} \tag{48}$$

According to the condition in Eq. (12) and (13), the $0 < \kappa_t(x)|x| < 2b^{(t-1)}$. Then, $|\kappa_t(x)|x| - b^{(t-1)}| < b^{(t-1)}$, there must $\exists \eta < 1$, so that $|\kappa_t(x)|x| - b^{(t-1)}| \leq \eta b^{(t-1)} < b^{(t-1)}$. Substituting it to Eq. (48), there is

$$\begin{aligned} |\Delta_g x_i^{(t+1)}| &\leq \mu(A) \sum_{j\in\mathcal{S}, j\neq i} |\Delta_g x_i^{(t)}| + (|x_i^{(t+1)}|\kappa_{t+1}(x^{(t+1)}) - b^{(t)})\mathrm{sign}(x_i^{(t+1)}) \\ &\leq \mu(A) \sum_{j\in\mathcal{S}, j\neq i} |\Delta_g x_i^{(t)}| + \eta b^{(t)}. \end{aligned} \tag{49}$$

Accumulate all the $|\Delta_g x_i^{(t+1)}|$ with all $i \in \mathcal{S}$, and define $s^{(t)} = |\mathrm{supp}(x^{(t)})|$ as the number of non-zeros elements in $x^{(t)}$ there is

$$\begin{aligned} \|\Delta_g x^{(t+1)}\|_1 &\leq \sum_{i\in\mathcal{S}} \mu(A) \sum_{j\in\mathcal{S}, j=i} |\Delta_g x_i^{(t)}| + (s^{(t+1)}\eta + (|\mathcal{S}| - s^{(t+1)}))b^{(t)} \\ &\leq \sum_{i\in\mathcal{S}} \mu(A) \sum_{j\in\mathcal{S}, j=i} |\Delta_g x_i^{(t)}| + (s^{(t+1)}\eta + (|\mathcal{S}| - s^{(t+1)}))b^{(t)} \\ &\leq (|\mathcal{S}| - 1)\mu(A)\|\Delta_g x^{(t)}\|_1 + (s^{(t+1)}\eta + (|\mathcal{S}| - s^{(t+1)}))\mu(A) \sup_{x_s} \|\Delta_g x^{(t)}\|_1. \end{aligned} \tag{50}$$

Take the supremum of Eq. (50), let $c_{t+1} = \log((2|\mathcal{S}| - 1 - s^{(t+1)}(1 - \eta))\mu(A))$ there is

$$\begin{aligned} \sup_{x_s} \|\Delta_g x^{(t+1)}\|_1 &\leq (|\mathcal{S}| - 1)\mu(A) \sup_{x_s} \|\Delta_g x^{(t)}\|_1 + (s^{(t+1)}\eta + (|\mathcal{S}| - s^{(t+1)}))\mu(A) \sup_{x_s} \|\Delta_g x^{(t)}\|_1 \\ &\leq (|\mathcal{S}| - 1 + s^{(t+1)}(1 - \eta)) \sup_{x_s} \|\Delta_g x^{(t)}\|_1 \\ &\leq \exp(c_{t+1}) \sup_{x_s} \|\Delta_g x^{(t)}\|_1 \\ &\leq \exp(\sum_{i=1}^{t+1} c_i) \sup_{x_s} \|\Delta_g x^{(0)}\|_1 \leq \exp(\sum_{i=1}^{t+1} c_i) sB. \end{aligned} \tag{51}$$

For the last layer (t-th layer), from Eq. (43), we have

$$\begin{aligned} x_i^{(t)} &= ((I - U^{(t)}A)(\Delta_g x^{(t)}))_i - \mu(A)\sup_{x_s}\|\Delta_g x^{(t)}\|_1 h(x_i^{(t+1)}) + (x_s)_i, \\ x_i^{(t)} - (x_s)_i &= ((I - U^{(t)}A)(\Delta_g x^{(t)}))_i - \mu(A) h(x_i^{(t+1)})\sup_{x_s}\|\Delta_g x^{(t)}\|_1, \\ |x_i^{(t)} - (x_s)_i| &\le |((I - U^{(t)}A)(\Delta_g x^{(t)}))_i| + \mu(A)\sup_{x_s}\|\Delta_g x^{(t)}\|_1. \end{aligned} \tag{52}$$

Using almost the same process in Eq. (25) and Eq. (26), we could deduce Eq. (52) that

$$\begin{aligned} \|x^{(t)} - x_s\|_1 &\le (2|\mathcal{S}| - 1)\mu(A)\sup_{x_s}\|\Delta_g x^{(t-1)}\|_1 \\ &\le (2|\mathcal{S}| - 1)\mu(A)\exp(\sum_{i=1}^{t-1} c_i) sB \\ &\le \exp(\sum_{i=1}^{t-1} c_i + c) sB, \end{aligned} \tag{53}$$

where $c = \log((2s-1)\mu(A))$.

Set $t_0 = \lceil \log(\frac{sB}{\|x_s\|_1}) / \log(\frac{1}{(2s-1)\mu(A)}) \rceil$. As $c_i \le c$, there will be $\|x^{(t)} - x_s\|_1 \le \exp(ct)sB$. When $i \ge t_0$, $\exp(ci)sB \le \|x_s\|_1$, $\|x^{(t)} - x_s\|_1 \le \|x_s\|_1$, then the $s^{(i)} = |\mathrm{supp}(x^{(i)})| > 0$, $c_i = \log(2|\mathcal{S}| - 1 + s^{(i)}(1-\eta))\mu(A) < \log((2s-1)\mu(A))$.

In conclusion, from Eq. (53), there is

$$\|x^{(t)} - x_s\|_2 \le \|x^{(t)} - x_s\|_1 \le \exp(\sum_{i=1}^{t-1} c_i + c) sB, \tag{54}$$

where $c = \log((2s-1)\mu(A))$, $c_i = c$ when $i < t_0$, and $c_i < c$ when $i \ge t_0$. □

A.4 PROOF OF THEOREM 3

Proof. For the t-th layer of the LISTA, according to the Eq. (22), we have

$$x_i^{(t+1)} = ((I - U^{(t)}A)(x^{(t)} - x_s))_i + (x_s)_i - b^{(t)} h(x_i^{(t+1)}). \tag{55}$$

As we have remove the no false positive assumption, $\mathrm{supp}(x_i^{(t)}) \nsubseteq \mathcal{S}$. Define $\mathcal{S}^{(t)}$ as $\forall i \in \mathcal{S}^{(t)}$ satisfies $i \in \mathrm{supp}(x_i^{(t)})$ but $i \notin \mathcal{S}$.

If $i \in \mathcal{S}$, from Eq. (55), we can deduce the same formulation as Eq. (24):

$$|x_i^{(t+1)} - (x_s)_i| \le \sum_{j \in \mathrm{supp}(x^{(t)})} (I - U^{(t)}A)_{ij}(x_j^{(t)} - (x_s)_j) + b^{(t)}. \tag{56}$$

If $i \in \mathcal{S}^{(t+1)}$,then the $(x_s)_i = 0$. From Eq. (55), there is

$$\begin{aligned} x_i^{(t+1)} &= ((I - U^{(t)}A)(x^{(t)} - x_s))_i - b^{(t)} h(x_i^{(t+1)}). \\ x_i^{(t+1)} &= ((I - U^{(t)}A)(x^{(t)} - x_s))_i - b^{(t)} \mathrm{sign}(x_i^{(t+1)}). \end{aligned} \tag{57}$$

Multiply $\mathrm{sign}(x_i^{(t+1)})$ on Eq. (57), we have

$$\begin{aligned} |x_i^{(t+1)}| &= ((I - U^{(t)}A)(x^{(t)} - x_s))_i \mathrm{sign}(x_i^{(t+1)}) - b^{(t)}, \\ |x_i^{(t+1)}| + b^{(t)} &= ((I - U^{(t)}A)(x^{(t)} - x_s))_i \mathrm{sign}(x_i^{(t+1)}), \\ (|x_i^{(t+1)}| + b^{(t)})\mathrm{sign}(x_i^{(t+1)}) &= ((I - U^{(t)}A)(x^{(t)} - x_s))_i, \end{aligned} \tag{58}$$

which means $((I - U^{(t)}A)(x^{(t)} - x_s))_i$ have the same sign with $b^{(t)}\text{sign}(x_i^{(t+1)})$. From the Eq.(57), there is

$$\begin{aligned} x_i^{(t+1)} - (x_s)_i = x_i^{(t+1)} &= \text{sign}(x_i^{(t+1)})(\sum_{j \in \text{supp}(x^{(t)})} (I - U^{(t)}A)_{ij}(x_j^{(t)} - (x_s)_j) - b^{(t)}) \\ |x_i^{(t+1)} - (x_s)_i = x_i^{(t+1)}| &\leq \sum_{j \in \text{supp}(x^{(t)})} |(I - U^{(t)}A)_{ij}(x_j^{(t)} - (x_s)_j)| - b^{(t)}. \end{aligned} \tag{59}$$

Accumulate all the $|x_i^{(t+1)} - (x_s)_i|$ with $i \in \text{supp}(x^{(t+1)}) = \mathcal{S}^{(t+1)} + \mathcal{S}$, there is

$$\begin{aligned} \|x^{(t+1)} - x_s\|_1 &\leq \sum_{i \in \mathcal{S}^{(t+1)} + \mathcal{S}} \sum_{j \in \text{supp}(x^{(t)})} |(I - U^{(t)}A)_{ij}(x_j^{(t)} - (x_s)_j)| + (|\mathcal{S}| - |\mathcal{S}^{(t+1)}|)|b^{(t)}|, \\ &\leq |\mathcal{S}^{(t+1)}| + |\mathcal{S}|\mu(A)\|x^{(t)} - x_s\|_1 + (|\mathcal{S}| - |\mathcal{S}^{(t+1)}|)b^{(t)}. \end{aligned} \tag{60}$$

Substitute the $b^{(t)} = \omega_{t+1}(k_{t+1}|\Theta)\mu_A \sup_{x_s} \|x^{(t)} - x_s\|_1$ into Eq. (60), and take its supremum:

$$\begin{aligned} \sup_{x_s} \|x^{(t+1)} - x_s\|_1 \leq\ & (|\mathcal{S}^{(t+1)}| + |\mathcal{S}|)\mu(A) \sup_{x_s} \|x^{(t)} - x_s\|_1 \\ & + (|\mathcal{S}| - |\mathcal{S}^{(t+1)}|)\omega_{t+1}(k_{t+1}|\Theta)\mu_A \sup_{x_s} \|x^{(t)} - x_s\|_1 \\ & \leq (|\mathcal{S}^{(t+1)}| + |\mathcal{S}| + (|\mathcal{S}| - |\mathcal{S}^{(t+1)}|)\omega_{t+1}(k_{t+1}|\Theta)) \\ & \mu(A) \sup_{x_s} \|x^{(t)} - x_s\|_1. \end{aligned} \tag{61}$$

Let $c_{t+1}^* = \log((|\mathcal{S}^{(t+1)}| + |\mathcal{S}| + (|\mathcal{S}| - |\mathcal{S}^{(t+1)}|)\omega_{t+1}(k_{t+1}|\Theta))\mu(A))$, and substitute it to Eq. (61),

$$\begin{aligned} \sup_{x_s} \|x^{(t+1)} - x_s\|_1 &\leq \exp(c_{t+1}^*) \sup_{x_s} \|x^{(t)} - x_s\|_1 \\ &\leq \exp(\sum_{i=1}^{t+1} c_i^*) \sup_{x_s} \|x^{(0)} - x_s\|_1 \\ &\leq \exp(\sum_{i=1}^{t+1} c_i^*) sB. \end{aligned} \tag{62}$$

According to the definition of $\omega_t(\cdot)$, $b^{(t)} = \omega_{t+1}(k_{t+1}|\Theta)\mu_A \sup_{x_s} \|x^{(t)} - x_s\|_1$, so that the number of false positive is less or equal than k_{t+1}, i.e. $|\mathcal{S}^{(t+1)}| \leq k_{t+1}$. As Assumption of ω_{t+1},$0 < \exists k_0^{t+1} < s$, $\omega_{k+1}(k_0^{t+1}|\Theta) < 1 - 1/(s - k_0^{t+1})^3$. Let $k_{t+1} = k_0^{t+1}$,

$$\begin{aligned} c_{t+1}^* &= \log((|\mathcal{S}^{(t+1)}| + |\mathcal{S}| + (|\mathcal{S}| - |\mathcal{S}^{(t+1)}|)\omega_{t+1}(k_{t+1}|\Theta))\mu(A)) \\ &= \log((|\mathcal{S}^{(t+1)}|(1 - \omega_{t+1}(k_{t+1}|\Theta)) + |\mathcal{S}|(1 + \omega_{t+1}(k_{t+1}|\Theta)))\mu(A)) \\ &\leq \log(k_{t+1}(1 - \omega_{t+1}(k_{t+1}|\Theta)) + s(1 + \omega_{t+1}(k_{t+1}|\Theta))\mu(A) \\ &< \log((k_{t+1}(\frac{1}{s - k_{t+1}}) + s(2 - \frac{1}{s - k_{t+1}}))\mu(A)) \\ &= \log((\frac{k_{t+1} + s(2s - 2k_{t+1} - 1)}{s - k_{t+1}})\mu(A)) \\ &= \log((2s - 1)\mu(A)). \end{aligned} \tag{63}$$

In conclusion, the l_2 error bound of the t-th layer of LISTA is

$$\|x^{(t)} - x_s\|_2 \leq \|x^{(t)} - x_s\|_1 \leq \sup_{x_s} \|x^{(t)} - x_s\|_1 \leq sB \exp(\sum_{i=1}^{t} c_i^*), \tag{64}$$

where $c_i^* < \log((2s - 1)\mu(A))$.

□

[3]According to the definition of ω_t, ω_t must be a monotonic decreasing function and $\omega_t(k|\Theta) < 1$ when $k > 0$.

A.5 Proof of Theorem 4

Proof. For the t-th layer given in Eq. (10), according to Eq. (46),

$$x^{(t+1)} = ((I - U^{(t)}A)(\Delta_g x^{(t)})) - b^{(t)}h(x^{(t+1)}) + x_s, \tag{65}$$

and

$$\Delta_g x^{(t+1)} = (I - U^{(t)}A)(\Delta_g x^{(t)}) - b^{(t)}h(x^{(t+1)}) + x^{(t+1)}\kappa_t(x^{(t+1)}). \tag{66}$$

As the no false positive is not fit for $x^{(t)}$, $x^{(t)} \nsubseteq \mathcal{S}$. We still define $\mathcal{S}^{(t)}$ as $\forall i \in \mathcal{S}^{(t)}$ satisfies $i \in \text{supp}(x_i^{(t)})$ but $i \notin \mathcal{S}$ and define $\mathbb{S}^{(t)}$ as $\forall i \in \mathbb{S}^{(t)}$ satisfies $i \in \mathcal{S}$ and $i \in \text{supp}(x_i^{(t)})$.

For $i \in \mathbb{S}^{(t+1)}$, $x_i^{(t+1)} \neq 0$, and $(x_s)_i \neq 0$. Substitute the form of k_t into i-th equation of Eq. (66):

$$\begin{aligned} \Delta_g x_i^{(t+1)} &= ((I - U^{(t)}A)(\Delta_g x^{(t)}))_i - b^{(t)}\text{sign}(x^{(t+1)}) + \mu_{t+1}b^{(t)}\text{sign}(x_i^{(t+1)}) \\ &= ((I - U^{(t)}A)(\Delta_g x^{(t)}))_i - (1-\mu_{t+1})b^{(t)}\text{sign}(x^{(t+1)}), \\ |\Delta_g x_i^{(t+1)}| &\leq |((I - U^{(t)}A)(\Delta_g x^{(t)}))_i| + (1-\mu_{t+1})b^{(t)}. \end{aligned} \tag{67}$$

For $i \notin \mathbb{S}^{(t+1)}$ but $i \in \mathcal{S}$, $x_i^{(t+1)} = 0$, and $(x_s)_i \neq 0$. The i-th equation of Eq. (66) is

$$\begin{aligned} \Delta_g x_i^{(t+1)} &= ((I - U^{(t)}A)(\Delta_g x^{(t)}))_i - b^{(t)}\text{sign}(x^{(t+1)}) \\ |\Delta_g x_i^{(t+1)}| &\leq |((I - U^{(t)}A)(\Delta_g x^{(t)}))_i| + b^{(t)}. \end{aligned} \tag{68}$$

For $i \in \mathcal{S}^{(t+1)}$, $(x_s)_i = 0$.

$$\Delta_g x_i^{(t+1)} = x_i^{(t+1)} g_t(x_i^{(t+1)}) = ((I - U^{(t)}A)(\Delta_g x^{(t)}))_i - (1-\mu_{t+1})b^{(t)}\text{sign}(x^{(t+1)}), \tag{69}$$

Multiply $\text{sign}(x_i^{(t+1)})$ on Eq. (69)

$$\begin{aligned} |x_i^{(t+1)} g_t(x_i^{(t+1)})| &= ((I - U^{(t)}A)(\Delta_g x^{(t)}))_i \text{sign}(x^{(t+1)}) - (1-\mu_{t+1})b^{(t)}, \\ |x_i^{(t+1)} g_t(x_i^{(t+1)})| + (1-\mu_{t+1})b^{(t)} &= ((I - U^{(t)}A)(\Delta_g x^{(t)}))_i \text{sign}(x^{(t+1)}), \end{aligned} \tag{70}$$

which means the $((I - U^{(t)}A)(\Delta_g x^{(t)}))_i$ should have the same sign with $\text{sign}(x^{(t+1)})$, i.e.

$$\begin{aligned} \Delta_g x_i^{(t+1)} &= \text{sign}(x^{(t+1)})(|((I - U^{(t)}A)(\Delta_g x^{(t)}))_i| - |(1-\mu_{t+1})b^{(t)}|) \\ |\Delta_g x_i^{(t+1)}| &\leq |((I - U^{(t)}A)(\Delta_g x^{(t)}))_i| - (1-\mu_{t+1})b^{(t)}. \end{aligned} \tag{71}$$

Accumulate all the $|\Delta_g x_i^{(t+1)}|$ with $i \in \text{supp}(x^{(t+1)})$, there is

$$\begin{aligned} \|\Delta_g x^{(t+1)}\|_1 &= \sum_{i \in \mathcal{S}^{(t+1)}, i \in \mathbb{S}^{(t+1)}, i \in \{\mathbb{S} - \mathbb{S}^{(t+1)}\}} |\Delta_g x_i^{(t+1)}| \\ &\leq \sum_{i \in \text{supp}(x^{(t+1)})} |((I - U^{(t)}A)(\Delta_g x^{(t)}))_i| + ((|\mathbb{S}^{(t+1)}| - |\mathcal{S}^{(t+1)}|)(1-\mu_{t+1}) \\ &\quad + (\mathcal{S} - |\mathbb{S}^{(t+1)}|))b^{(t)} \\ &\leq (|\mathcal{S}| + |\mathcal{S}^{(t+1)}|)\mu(A)\|\Delta_g x^{(t)}\|_1 + ((|\mathbb{S}^{(t+1)}| - |\mathcal{S}^{(t+1)}|)(1-\mu_{t+1}) \\ &\quad + (\mathcal{S} - |\mathbb{S}^{(t+1)}|))b^{(t)}. \end{aligned} \tag{72}$$

Substitute the $b^{(t)} = \omega_{t+1}(k_{t+1}|\Theta)\mu(A)\sup_{x_s}\|\Delta_g x^{(t)}\|_1$ into Eq. (72). Take the supremum of Eq. (72):

$$\begin{aligned} \sup_{x_s} \|\Delta_g x^{(t+1)}\|_1 \leq\ & (|\mathcal{S}| + |\mathcal{S}^{(t+1)}|)\mu(A)\|\sup_{x_s}\Delta_g x^{(t)}\|_1 + ((|\mathbb{S}^{(t+1)}| - |\mathcal{S}^{(t+1)}|)(1-\mu_{t+1}) \\ & + (\mathcal{S} - |\mathbb{S}^{(t+1)}|))\omega_{t+1}(k_{t+1}|\Theta)\mu(A)\sup_{x_s}\|\Delta_g x^{(t)}\|_1 \\ \leq\ & (|\mathcal{S}| + |\mathcal{S}^{(t+1)}| + ((|\mathbb{S}^{(t+1)}| - |\mathcal{S}^{(t+1)}|)(1-\mu_{t+1}) \\ & + (\mathcal{S} - |\mathbb{S}^{(t+1)}|))\omega_{t+1}(k_{t+1}|\Theta))\mu(A)\sup_{x_s}\|\Delta_g x^{(t)}\|_1. \end{aligned} \tag{73}$$

Let $c'_{t+1} = \log((|\mathcal{S}| + |\mathcal{S}^{(t+1)}| + ((|\mathbb{S}^{(t+1)}| - |\mathcal{S}^{(t+1)}|)(1 - \mu_{t+1}) + (\mathcal{S} - |\mathbb{S}^{(t+1)}|))\omega_{t+1}(k_{t+1}|\Theta))\mu(A))$. The $\sup_{x_s} \|\Delta_g x^{(t+1)}\|_1$ satisfies

$$\begin{aligned}\sup_{x_s} \|\Delta_g x^{(t+1)}\|_1 &\leq \exp(c'_{t+1}) \sup_{x_s} \|\Delta_g x^{(t)}\|_1 \\ &\leq \exp(\sum_{i=1}^{t+1} c'_i) \sup_{x_s} \|\Delta_g x^{(0)}\|_1 \\ &\leq \exp(\sum_{i=1}^{t+1} c'_i) sB.\end{aligned} \tag{74}$$

Consider about the relationship between $x^{(t+1)} - x_s$ and $\Delta_g x^{(t)}$:

If $i \in \mathcal{S}^{(t+1)}$,$(x_s)_i = 0$, and $x_i^{(t+1)} \neq 0$. The i-th equation in Eq. (65) is

$$x_i^{(t+1)} - (x_s)_i = ((I - U^{(t)}A)(\Delta_g x^{(t)}))_i - b^{(t)}\text{sign}(x_i^{(t+1)}). \tag{75}$$

According to the similar analyses in previous, the sign of $((I - U^{(t)}A)(\Delta_g x^{(t)}))_i$ is the same as $\text{sign}(x_i^{(t+1)})$, $x_i^{(t+1)} - (x_s)_i$ satisfies

$$\begin{aligned}x_i^{(t+1)} - (x_s)_i &= (|((I - U^{(t)}A)(\Delta_g x^{(t)}))_i| - b^{(t)})\text{sign}(x_i^{(t+1)}), \\ |x_i^{(t+1)} - (x_s)_i| &= |((I - U^{(t)}A)(\Delta_g x^{(t)}))_i| - b^{(t)}.\end{aligned} \tag{76}$$

$i \in \mathcal{S}$, the $x_i^{(t+1)} - (x_s)_i$ satisfies

$$|x_i^{(t+1)} - (x_s)_i| \leq |((I - U^{(t)}A)(\Delta_g x^{(t)}))_i| + b^{(t)}. \tag{77}$$

Accumulate all the $|x_i^{(t+1)} - (x_s)_i|$ with $i \in \text{supp}(x^{(t+1)})$, there is

$$\begin{aligned}\|x^{(t+1)} - x_s\|_1 &\leq \sum_{i \in \text{supp}(x^{(t+1)})} |((I - U^{(t)}A)(\Delta_g x^{(t)}))_i| + (|\mathcal{S}| - \mathcal{S}^{(t+1)})|b^{(t)}| \\ &\leq (|\mathcal{S}| + \mathcal{S}^{(t+1)})\mu(A)\|\Delta_g x^{(t)}\|_1 + (|\mathcal{S}| - \mathcal{S}^{(t+1)})b^{(t)}.\end{aligned} \tag{78}$$

Substitute the $b^{(t)} = \omega_{t+1}(k_{t+1}|\Theta)\mu(A) \sup_{x_s} \|\Delta_g x^{(t)}\|_1$ into Eq. (78), take the supremum of $\|x^{(t+1)} - x_s\|_1$ and $\|\Delta_g x^{(t)}\|_1$:

$$\begin{aligned}\sup_{x_s} \|x^{(t+1)} - x_s\|_1 &\leq (|\mathcal{S}| + \mathcal{S}^{(t+1)})\mu(A) \sup_{x_s} \|\Delta_g x^{(t)}\|_1 \\ &\quad + (|\mathcal{S}| - \mathcal{S}^{(t+1)})\omega_{t+1}(k_{t+1}|\Theta) \sup_{x_s} \|\Delta_g x^{(t)}\|_1 \\ &\leq (|\mathcal{S}| + \mathcal{S}^{(t+1)} + (|\mathcal{S}| - \mathcal{S}^{(t+1)})\omega_{t+1}(k_{t+1}|\Theta))\mu(A) \sup_{x_s} \|\Delta_g x^{(t)}\|_1.\end{aligned} \tag{79}$$

Let $c'^* = \log((|\mathcal{S}| + \mathcal{S}^{(t)} + (|\mathcal{S}| - \mathcal{S}^{(t)})\omega_t(k_t|\Theta))\mu(A))$. Substitute Eq. (74) into Eq. (79) The l_2 error bound of LISTA with gain gate should be

$$\begin{aligned}\|x^{(t)} - x_s\|_2 \leq \|x^{(t)} - x_s\|_1 &\leq \sup_{x_s} \|x^{(t)} - x_s\|_1 \\ &\leq \exp(\sum_{i=1}^{t-1} c'_i + c'^*) sB\end{aligned} \tag{80}$$

Let $t_0 = \lceil \log(\frac{sB}{\sigma}) / \log(\frac{1}{(2s-1)\mu(A)}) \rceil$. When $i < t_0$, as $\mu_i = 0$, $c'_i = \log((|\mathcal{S}| + |\mathcal{S}^{(i)}| + (\mathcal{S} - \mathcal{S}^{(i)})\omega_{i+1}(k_{i+1}|\Theta))\mu(A))$. According to the prove mean process of Theorem 4, let $k_i = k_0^i$, $0 < k_i < s$, and $c'_i = c_i^* < \log(\frac{1}{(2s-1)\mu(A)})$.

When $i \geq t_0$, $\|x^{(i)} - x_s\|_1 < sB \exp(ci) \leq \sigma$. As the minimal absolute value of x_s is less or equal than σ, $\mathbb{S}^{(t)} = \mathcal{S}$. When $k_i = |\mathcal{S}^{(i)}| = 0$, $c'_i = |\mathcal{S}| + |\mathcal{S}|(1 - \mu_i)$. As $1 - \omega_i(s|\Theta) < \mu_i \leq 1$, $1 - \mu_i < \omega_i(s|\Theta)$. Recall the form the $c_i^* \leq \log((|\mathcal{S}| + |\mathcal{S}^{(i)}| + (\mathcal{S} - \mathcal{S}^{(i)})\omega_i(k_i|\Theta)\mu(A))$. As $|\mathcal{S}| + |\mathcal{S}|(1 - \mu_i) < |\mathcal{S}| + |\mathcal{S}|\omega_i(s|\Theta) < |\mathcal{S}| + |\mathcal{S}|\omega_i(k_i|\Theta) \leq |\mathcal{S}| + |\mathcal{S}^{(i)}| + (|\mathcal{S}| - |\mathcal{S}^{(i)}|)\omega_i(k_i|\Theta)$. The $c'_i \leq c_i^*$.

□

A.6 Proof of Proposition 2

Recall that the update rule of ISTA is $x^{(0)} = \mathbf{0}$ and

$$x^{(t+1)} = s_{\lambda/\gamma}(x^{(t)} - \nabla_x f(x^{(t)})/\gamma). \tag{81}$$

We have the following theorem which analyzes the update rule of ISTA and

$$\eta^* := \arg\min_{\eta} f(\eta(x^{(t+1)} - x^{(t)}) + x^{(t)}, y) + \lambda \|\eta(x^{(t+1)} - x^{(t)}) + x^{(t)}\|_1. \tag{82}$$

Proof. According to the analysis in Section 2 in the main paper, $x^{(t+1)}$ is the solution of minimizing the upper bound $U(x)$,

$$U(x) := f(x^{(t)}, y) + (x - x^{(t)})\nabla_x f(x^{(t)}) + \frac{\gamma}{2}\|x - x^{(t)}\|^2 + \lambda r(x). \tag{83}$$

The sub-gradient of $U(x)$ is

$$\partial_x U(x) = \nabla_x f(x^{(t)}) + \gamma(x - x^{(t)}) + \lambda \partial_x r(x). \tag{84}$$

As the $x^{(t+1)}$ is the optimal solution to minimizing Eq. (83), $\partial_x U(x^{(t+1)})$ satisfies

$$0 \in \partial_x U(x^{(t+1)}) = \nabla_x f(x^{(t)}) + \gamma(x^{(t+1)} - x^{(t)}) + \lambda \partial_x r(x^{(t+1)}), \tag{85}$$

where $r(x) = \|x\|_1$. According to the definition of the sub-gradient, $(\partial_x r(x))_i \in [-1, 1]$ when $x_i = 0$, $(\partial_x r(x))_i = -1$ when $x_i < 0$, and $(\partial_x r(x))_i = 1$ when $x_i > 0$.

From the Eq. (85), there must exists $r_1 \in r(x)$ such that

$$\nabla_x f(x^{(t)}) + \gamma(x^{(t+1)} - x^{(t)}) + \lambda r_1 = 0, \tag{86}$$

where $(r_1)_i = 1$ if $x_i^{(t+1)} > 0$, $(r_1)_i = -1$ if $x_i^{(t+1)} < 0$, and $-1 \leq (r_1)_i \leq 1$ if $x_i^{(t+1)} < 0$.

According to the definition of η^* in Eq. (82), we define a new function $\theta(\eta)$ as

$$\theta(\eta) = f(\eta(x^{(t+1)} - x^{(t)}) + x^{(t)}, y) + \lambda\|\eta(x^{(t+1)} - x^{(t)}) + x^{(t)}\|_1. \tag{87}$$

Notice that $\theta(\eta)$ is the line search function of $f(x, y) + \lambda\|x\|_1$. According to the law of convex optimization, as $f(x, y) + \lambda\|x\|_1$ is a convex function, the $\theta(\eta)$ must be also a convex function about η. The sub-gradient of $\theta(\eta)$ is

$$\begin{aligned}\partial_x \theta(\eta) = (x^{(t+1)} - x^{(t)})^T \nabla_x f(\eta(x^{(t+1)} - x^{(t)}) + x^{(t)}) +\\ \lambda(x^{(t+1)} - x^{(t)})^T \partial_x r(\eta(x^{(t+1)} - x^{(t)}) + x^{(t)}).\end{aligned} \tag{88}$$

The η^* actually is the value to minimize $\theta(\eta)$ in Eq. (87). There must be

$$0 \in \partial_x \theta(\eta^*). \tag{89}$$

From Eq. (88), the sub-gradient function of $\theta(\eta)$ when $\eta = 1$ is

$$\begin{aligned}&\partial_\eta \theta(1)\\ =&(x^{(t+1)} - x^{(t)})^T(\nabla_x f(x^{(t+1)}) + \lambda\partial_x r(x^{(t+1)}))\\ =&(x^{(t+1)} - x^{(t)})^T(\nabla_x f(x^{(t+1)}) - \nabla_x f(x^{(t)}) + \nabla_x f(x^{(t)})) + \lambda\partial_x r(x^{(t+1)}))\\ =&(x^{(t+1)} - x^{(t)})^T(\nabla_x^2 f(\zeta)(x^{(t+1)} - x^{(t)}) + \nabla_x f(x^{(t)}) + \lambda\partial_x r(x^{(t+1)})),\end{aligned} \tag{90}$$

where $\zeta \in \mathbb{R}^n$. Substitute $\nabla_x f(x^{(t)})$ in Eq. (86) into Eq. (90), the $\partial_\eta \theta(1)$ is

$$\begin{aligned}&\partial_\eta \theta(1)\\ =&(x^{(t+1)} - x^{(t)})^T(\nabla_x^2 f(\zeta)(x^{(t+1)} - x^{(t)}) - \gamma(x^{(t+1)} - x^{(t)}) - \lambda r_1 + \lambda\partial_x r(x^{(t+1)})\\ =&(x^{(t+1)} - x^{(t)})^T((\nabla_x^2 f(\zeta) - \gamma I)(x^{(t+1)} - x^{(t)}) + \lambda(\partial r(x^{(t+1)}) - r_1))\\ =&(x^{(t+1)} - x^{(t)})^T(\nabla_x^2 f(\zeta) - \gamma I)(x^{(t+1)} - x^{(t)}) +\\ &\lambda\sum_i (x_i^{(t+1)} - x_i^{(t)})((\partial r(x^{(t+1)}))_i - (r_1)_i)\\ =&(x^{(t+1)} - x^{(t)})^T(\nabla_x^2 f(\zeta) - \gamma I)(x^{(t+1)} - x^{(t)}) +\\ &\lambda\sum_i (x_i^{(t+1)} - x_i^{(t)})((r)_i - (r_1)_i),\end{aligned} \tag{91}$$

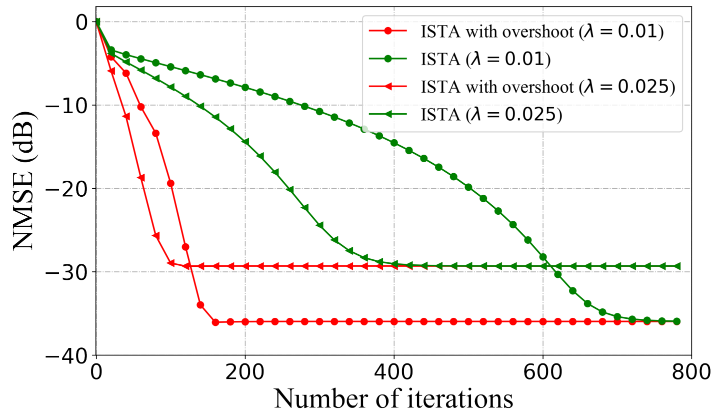


Figure 7: Experimental results validating our Proposition 2. It can be observed that the update of ISTA "lags behind".

where $(r)_i \in (\partial r(x^{(t+1)}))_i$. According to the properties of convex function and the sub-gradient, assume $\eta^* < 1$, s.t. $0 \in \partial_\eta \theta(\eta^*)$, $\forall r_\theta \in \partial_\eta \theta(1)$, there will be $r_\theta \geq 0$. However, as $r_1 \in \partial r(x^{(t+1)})$, substitute $r = r_1 \in \partial r(x^{(t+1)})$ into Eq. (91). The corresponding element in sub-gradient when $r = r_1$ is $r_\theta = (x^{(t+1)} - x^{(t)})^T(\nabla_x^2 f(\zeta) - \gamma I)(x^{(t+1)} - x^{(t)}) \in \partial_\eta \theta(1)$. According to given condition $\gamma I - \nabla_x^2 f(x) \succ 0$, $r_\theta < 0$, which is in contrast to $\eta^* < 1$. Therefore, the conclusion

$$\eta^* \geq 1 \tag{92}$$

is obtained.

Moreover, consider about the last term of Eq. (91), i.e.

$$\sum_i (x_i^{(t+1)} - x_i^{(t)})((\partial r(x^{(t+1)}))_i - (r_1)_i). \tag{93}$$

If $\text{supp}(x^{(t)}) \subset \text{supp}(x^{(t+1)})$, there are two situations about index i. 1) $i \in \text{supp}(x^{(t+1)})$, there will be $x_i^{(t+1)} \neq 0$ and $(\partial r(x^{(t+1)}))_i = (r_1)_i = \text{sign}(x_i^{(t+1)})$. 2) $i \notin \text{supp}(x^{(t+1)})$ and $i \notin \text{supp}(x^{(t)})$, there will be $x_i^{(t+1)} = x_i^{(t)} = 0$. Both conditions will make the term $(x_i^{(t+1)} - x_i^{(t)})((\partial r(x^{(t+1)}))_i - (r_1)_i)$ in Eq. (93) 0. Therefore, Eq. (93) is

$$\sum_i (x_i^{(t+1)} - x_i^{(t)})((\partial r(x^{(t+1)}))_i - (r_1)_i) = 0. \tag{94}$$

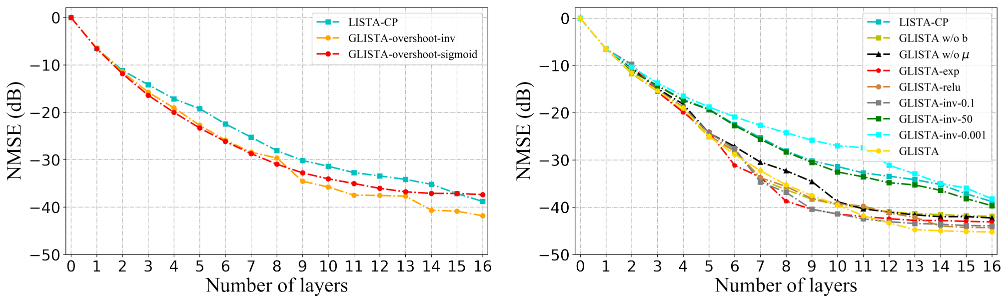
According to the given condition $\gamma I - \nabla_x^2 f(x) \succ 0$, $\partial_\eta \theta(1)$ should be a number but not a set and $\partial_\eta \theta(1) = (x^{(t+1)} - x^{(t)})^T(\nabla_x^2 f(\zeta) - \gamma I)(x^{(t+1)} - x^{(t)}) < 0$. As the $\theta(\eta)$ is convex function, there must be $\eta^* > 1$ because of $0 \in \partial_\eta \theta(\eta^*)$. The conclusion $\eta^* > 1$ is derived. □

B More Simulation Experiments

B.1 Sparse Coding with Gain Gates

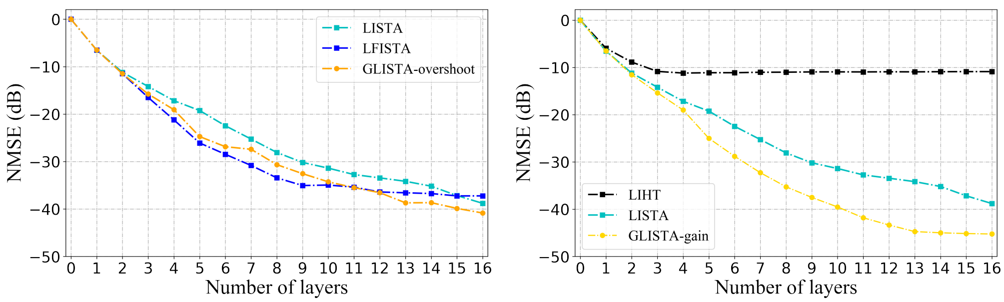
Validation of Proposition 2: Some more experimental results are given here due to the length limit of the main body of our paper. One might also be interested in our Proposition 2, hence we first conduct an experiment to confirm it. We adopt ISTA with an adaptive overshoot and compare it with the standard ISTA for sparse coding. The adaptation is obtained via enlarging the step size from 1.0 through backtracking line search (see section 7 for more details). Figure 7 demonstrates that our overshoot mechanism facilitates ISTA optimization, and such a result confirms Proposition 2.

Discussions about the overshoot and gain gates: It should be interesting to compare the performance of our gates with different expressions. We test LISTA with two different overshoot gate functions in Figure 8(a). The first one is a sigmoid-based function which has been shown in the



(a) (b)

Figure 8: Comparison of different (a) overshoot gate functions and (b) gain gate functions. The experiment is performed with SNR=40dB.



(a) (b)

Figure 9: Comparison of overshoot and gain gate with similar method.

main body of our paper, i.e.,

$$o_t(x^{(t)}, y|\Lambda_o^{(t)}) = 1 + a_o \sigma(W_o x^{(t)} + U_o y) \left| \sum_i y_i \right|, \tag{95}$$

in which a_o is a learnable parameter constrained to be non-negative and $\sigma(\cdot)$ is the sigmoid function, and the second one is

$$o_t(x^{(t)}, y|\Lambda_o^{(t)}) = 1 + \frac{a_o}{|\tilde{x}^{(t+1)} - x^{(t)}| + \epsilon}, \tag{96}$$

in which ϵ is a tiny positive constant introduced to avoid zero being divided. Both of the functions are incorporated into LISTA with their learnable parameters being shared among layers. It can be seen from Figure 8(a) that the acceleration in convergence and gain in final performance are obvious, just as expected.

For LISTA with our gain gates, one can check the results in Figure 8(b). It can be seen that if either the bias term or the μ_t term is removed, the performance of our gated LISTA degrades a lot. We also try different $f_t(\cdot)$ functions as mentioned, including a ReLU-based one and some possibly more nonlinear ones. See Figure 8(b) for a comparison. We confirm that gate functions whose outputs are relatively closer to the boundary condition may perform better. However, it is worth noting that when the outputs reach that boundary condition, the performance also degrades (see the LISTA-inv-ϵ results in Figure 8(b)), which confirms the necessity of the μ_t term in designing our gain gate functions. The results further suggest to adopt a combination of gate functions in practice. Specifically, we use the ReLU-based function for the first 10 layers and the inverse proportional function for deeper layers in our experiments comparing with the state-of-the-arts, and we directly call it GLISTA for convenience.

As mentioned in the main body of the paper, the overshoot gates is proposed do address insufficient step size, which is similar to the motivation of (L)FISTA. LIHT and support select can also be considered as special cases of our gain gates (by letting $\mu_t = 1$ in the inverse proportional function). We compare these similar methods with our overshoot and gain gates in Figure 9. It can be seen that when compared with LISTA, LFISTA converges faster in lower layers, and our overshoot gates

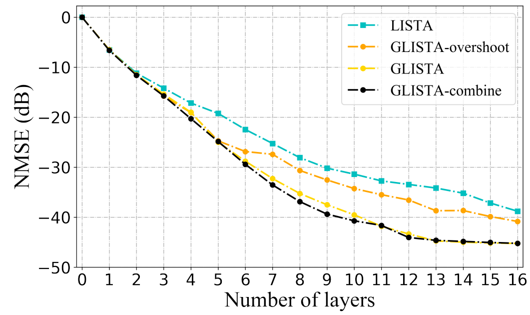
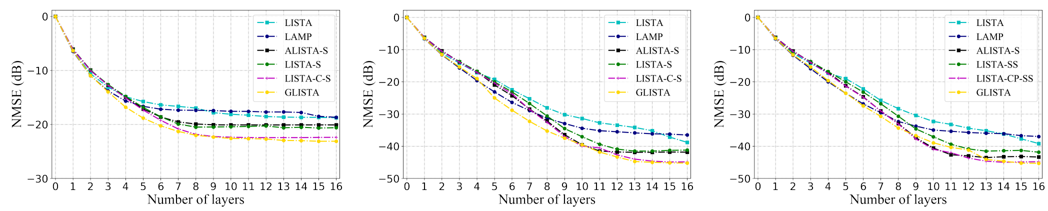


Figure 10: Comparison of the gain, overshoot, and their combinations.



(a) SNR=20dB (b) SNR=40dB (c) condition number=3

Figure 11: Comparison of sparse coding methods under different settings. Our GLISTA consistently outperforms the competitors in almost all test cases with different numbers of layers.

also show such advantage. When applying to deeper layers, LFISTA converges quite slow while the overshoot gates still perform well, which indicates that the time-varying property is beneficial in practice. LISTA with our gain gates is obviously better than LIHT as shown in Figure 9(b), and sufficient experimental results in the paper also prove that the gain gate outperforms support select (e.g., in LISTA-C-S and LISTA-S). We also test the combination of gain gates and overshoot gates, despite the fact that the mechanism with only gain gates is already good enough. See Figure 10 for an illustrative comparison. Apparently, when the overshoot gates are further incorporated, the convergence on lower layers become faster while the overall convergence is not affected much, leading to similar final performance when the model is very deep and superior performance when the model is relatively shallow.

Now we also give some sparse coding results under the less challenging settings on the noise level and the condition number in Figure 11. Compared with LISTA-CP, LAMP, LISTA-SS, and LISTA-CP-SS, our gated LISTA (GLISTA) performs remarkably better with ill-posed dictionary matrices and less noises. Table 3 and 4 report the statistical means of five runs using different methods. It can be seen that the improvement achieved by our GLISTA is significant.

Algorithm 1 ISTA with adaptive overshoot.

Input: The dictionary matrix A, an observation y, an initial step size $\eta_0 = 1.0$ for sparse coding, a step size $\tau = 1.05$ for line search, and a maximal number of iteration.

Output: output result

$x^{(0)} = 0$
for $t = 0, \cdots, K-1$ **do**
  $\tilde{x}^{(t)} = s_{\lambda/\gamma}((I - A^T A/\gamma)x^{(t-1)} + A^T y/\gamma)$
  $x_p = \tilde{x}^{(t)}$, $\eta = \eta_0 \tau$
  $x_c = \tau(\tilde{x}^{(t)} - x^{(t)}) - x^{(t)}$
  while $f_o(x_p, y) \geq f_o(x_c, y)$ **do**
    $x_p = x_c$, $\eta = \tau\eta$
    $x_c = \eta(\tilde{x}^{(t)} - x^{(t)}) - x^{(t)}$
  $x^{(t)} = x_p$
return x^{K-1}

C Adaptive Overshoot

We perform an adaptive overshoot in the experiment to confirm Proposition 2. The algorithm is summarized in Algorithm 1. Most of input variables are introduced in the main body of our paper and τ is given as the step size for performing line search. The whole algorithm procedure is very similar to the famous backtracking line search. The step size η for sparse coding is updated by τ until the objective function $f(x, y) + \lambda r(x)$ does not decrease any more.

Table 3: Comparison of the final NMSEs under different noise levels with $d = 16$. The condition number of the dictionary is not specifically constrained.

SNR	LISTA	LAMP	LISTA-S	LISTA-C-S	ALISTA-S	GLISTA (ours)
40	-38.72	-36.77	-41.99	-44.85	-41.86	**-45.22**
20	-18.65	-18.66	-20.64	-22.84	-20.00	**-23.08**
10	-9.42	-9.46	-9.84	-11.06	-9.04	**-11.41**

Table 4: Comparison of the final NMSEs under different condition numbers with $d = 16$. The noise level is chosen as SNR=40dB for all the tested condition numbers.

Con. num.	LISTA	LAMP	LISTA-S	LISTA-C-S	ALISTA-S	GLISTA (ours)
3	-39.03	-37.26	-43.12	-44.90	-43.88	**-45.33**
30	-29.65	-28.44	-32.30	-38.36	31.50	**-39.61**
100	-21.39	-22.23	-27.08	-27.94	27.10	**-34.07**

References

Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009.

Mark Borgerding, Philip Schniter, and Sundeep Rangan. Amp-inspired deep networks for sparse linear inverse problems. *IEEE Transactions on Signal Processing*, 65(16):4293–4308, 2017.

Xiaohan Chen, Jialin Liu, Zhangyang Wang, and Wotao Yin. Theoretical linear convergence of unfolded ista and its practical weights and thresholds. In *Advances in Neural Information Processing Systems*, pp. 9061–9071, 2018.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *EMNLP*, 2014.

Ingrid Daubechies, Michel Defrise, and Christine De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 57(11):1413–1457, 2004.

Bradley Efron, Trevor Hastie, Iain Johnstone, Robert Tibshirani, et al. Least angle regression. *The Annals of statistics*, 32(2):407–499, 2004.

Michael Elad and Michal Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image processing*, 15(12):3736–3745, 2006.

Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448, 2015.

Raja Giryes, Yonina C Eldar, Alex M Bronstein, and Guillermo Sapiro. Tradeoffs between convergence speed and reconstruction accuracy in inverse problems. *IEEE Transactions on Signal Processing*, 66(7):1676–1690, 2018.

Karol Gregor and Yann LeCun. Learning fast approximations of sparse coding. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, pp. 399–406. Omnipress, 2010.

Hao He, Bo Xin, Satoshi Ikehata, and David Wipf. From bayesian sparsity to gated recurrent nets. In *Advances in Neural Information Processing Systems*, pp. 5554–5564, 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

John R Hershey, Jonathan Le Roux, and Felix Weninger. Deep unfolding: Model-based inspiration of novel deep architectures. *arXiv preprint arXiv:1409.2574*, 2014.

Geoffrey Hinton, Li Deng, Dong Yu, George Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Brian Kingsbury, et al. Deep neural networks for acoustic modeling in speech recognition. *IEEE Signal processing magazine*, 29, 2012.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015.

Jialin Liu, Xiaohan Chen, Zhangyang Wang, and Wotao Yin. Alista: Analytic weights are as good as learned weights in lista. 2018.

Thomas Moreau and Joan Bruna. Understanding trainable sparse coding via factorization. In *In International Conference on Learning Representation*.

Jeremias Sulam, Aviad Aberdam, Amir Beck, and Michael Elad. On multi-layer basis pursuit, efficient algorithms and convolutional neural networks. *IEEE transactions on pattern analysis and machine intelligence*, 2019.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

Zhangyang Wang, Qing Ling, and Thomas S Huang. Learning deep l0 encoders. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

Bo Xin, Yizhou Wang, Wen Gao, David Wipf, and Baoyuan Wang. Maximal sparsity with deep networks? In *Advances in Neural Information Processing Systems*, pp. 4340–4348, 2016.

Jianchao Yang, John Wright, Thomas S Huang, and Yi Ma. Image super-resolution via sparse representation. *IEEE transactions on image processing*, 19(11):2861–2873, 2010.

Jian Zhang and Bernard Ghanem. Ista-net: Interpretable optimization-inspired deep network for image compressive sensing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1828–1837, 2018.